NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ARIES: AN ARCHITECTURAL IMPLEMENTATION OF A MULTI-CRITERION SPATIAL DECISION SUPPORT SYSTEM (SDSS)

by

Peter R. Falk

September 1997

Thesis Advisor — Daniel R. Dolk
Associate Advisor — Dale M. Courtney

Approved for Public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1997	Master's Thesis

4. TITLE AND SUBTITLE
ARIES: AN ARCHITECTURAL IMPLEMENTATION OF A MULTI-CRITERION SPATIAL DECISION SUPPORT SYSTEM (SDSS)

5. FUNDING NUMBERS

6. AUTHOR(S)
Peter R. Falk

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

This thesis describes a component-based methodology for developing a new class of systems called spatial decision support systems (SDSS). The methodology is presented within the context of the development of the ARIES (Army Reserve Installation Evaluation System) software application, an SDSS designed to evaluate and compare site desirability for Army Reserve unit locations. The ARIES SDSS consists of a flexible component-based architecture that seamlessly integrates a user interface, GIS, multi-criteria decision model with associated DSS, and data warehouse. To build the SDSS, the ARIES developers introduced a new architectural paradigm, undertaking a collaborative approach with U.S. Army Reserve Command (USARC) decision-makers to rapidly prototype ARIES using component-based technologies. The developers implemented several domain-specific architectures using a formalized proof-of-concept heuristic, Concept-to-Code (C2C), which conceptualizes user requirements in architectural terms, and migrating legacy data sources into a spatial data warehouse. C2C allowed the resultant ARIES application to be conceptualized initially in general terms, and then specialized architecturally around existing off the shelf components, as design requirements were collaboratively prototyped and implemented within the existing USARC information system infrastructure. C2C facilitated the complete development of a complex, map-based system and accompanying data warehouse in the span of a few months with a technical team of three analysts and programmers. Significant system performance gains resulted from instituting a Migration Architecture System (MARS) engine to extract and spatially enable relevant data sources for geographic querying. Additional performance enhancements were also obtained through the use of rapid, component-based development techniques.

14. SUBJECT TERMS
Software Architecture, Heuristic, Application Development, Spatial Decision Support System

15. NUMBER OF PAGES
294

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for Public release; distribution is unlimited.

ARIES: AN ARCHITECTURAL IMPLEMENTATION OF A MULTI-CRITERION SPATIAL DECISION SUPPORT SYSTEM (SDSS)

Peter R. Falk
Lieutenant, United States Navy
B.S., Illinois Institute of Technology, 1988

Submitted in partial fulfillment of the
requirements for the degree of

MASTERS OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September, 1997

Author: _____
Peter R. Falk

Approved by: _____
Daniel R. Dolk, Thesis Advisor

_____
Dale M. Courtney, Associate Advisor

_____
Reuben Harris, Chairman,
Department of Systems Management

ABSTRACT

This thesis describes a component-based methodology for developing a new class of systems called spatial decision support systems (SDSS). The methodology is presented within the context of the development of the ARIES (Army Reserve Installation Evaluation System) software application, an SDSS designed to evaluate and compare site desirability for Army Reserve unit locations. The ARIES SDSS consists of a flexible component-based architecture that seamlessly integrates a user interface, GIS, multi-criteria decision model with associated DSS, and data warehouse.

To build the SDSS, the ARIES developers introduced a new architectural paradigm, undertaking a collaborative approach with U.S. Army Reserve Command (USARC) decision-makers to rapidly prototype ARIES using component-based technologies. The developers implemented several domain-specific architectures using a formalized proof-of-concept heuristic, Concept-to-Code (C2C), which conceptualizes user requirements in architectural terms, and migrating legacy data sources into a spatial data warehouse.

C2C allowed the resultant ARIES application to be conceptualized initially in general terms, and then specialized architecturally around existing off the shelf components, as design requirements were collaboratively prototyped and implemented within the existing USARC information system infrastructure. C2C facilitated the complete development of a complex, map-based system and accompanying data warehouse in the span of a few months with a technical team of three analysts and programmers. Significant system performance gains resulted from instituting a Migration Architecture System (MARS) engine to extract and spatially enable relevant data sources for geographic querying. Additional performance enhancements were also obtained through the use of rapid, component-based development techniques.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND/OR ABBREVIATIONS

4GL	Fourth Generation Language
ARCOMS	Army Reserve Commands
AMSA	Area Maintenance Support Activity
API	Application Program Interface
ARGIS	Architecture Refinable Geographic Information System
ARIES	Army Reserve Installation Evaluation System
ARPERCEN	Army Reserve Personnel Center
ARU	Army Reserve Unit
ARU-DM	Army Reserve Unit Decision Model
C2C	Concept-To-Code
CBD	Component-based Development
CONUSA	Continental United States Army
COTS	Commercial-Off-The-Shelf
DOD	Department of Defense
DM	Decision-maker
DMW	Data Migration Warehouse
DSS	Decision Support System
ECS	Equipment Concentration Center
FacID	Facility Identification Code
FORSCOM	Forces Command
FSP	Force Support Package
GIS	Geographic Information System
GUI	Graphic User Interface
IG	Inspector General
IRR	Individual Ready Reserve
IS	Information Systems
IT	Information Technology
LAN	Local Area Network
LDW	Logical Decisions for Windows
MARS	Migration Architecture System
MOS	Military Occupational Specialty
NPS	Naval Postgraduate School
OCAR	Office of Chief Army Reserve

OLE	Object Linking and Embedding
RDBMS	Relational Database Management System
SDSS	Spatial Decision Support System
SQL	Structured Query Language
STSC	Software Technology Support Center
TPU	Troop Program Unit
U/I	User Interface
UIC	Unit Identification Code
USAF	United States Air Force
USARC	United States Army Reserve Center

ACKNOWLEDGEMENT

First and foremost I want to thank my family for allowing me the time to work on this project.

I would like to thank Professors Dolk and Thomas for giving me the opportunity to work on this project and for allowing me to take this topic in the direction that I wanted it to go.

Thank you to LCDR Dale M. Courtney for his assistance as second reader. His insights and guidance were very important to the final product of this thesis. He offered a objective opinion of someone outside the details of the ARIES project.

The development of the ARIES prototype would not have been possible without the generosity of Professor Barry Frew. He allowed the development team full use of a desktop computer that was essential to the projects success.

I. INTRODUCTION.

"There is nothing more difficult to take in hand, more perilous to conduct, or more uncertain in its success, than to take the lead in the introduction of a new order of things."
Niccolo Machiavelli, *The Prince,* 1532.

"We are living in an information age where almost everything is seen as technically possible given the commitment and the resources."
Martyn Jones,
"Information: It's Architecture and Management", 1997.

"We can no longer afford stovepiped systems. And it's not just an affordability issue, it's a warfighter issue."
General Ronald Griffith, Vice Chief of Staff of the Army, 1997.

A. THESIS OVERVIEW.

The primary purpose of the research conducted for this thesis is to investigate and document the architectural approaches used in developing the Army Reserve Installation Evaluation System (ARIES) spatial decision support system (SDSS) for the United States Army Reserve Command (USARC). The objective of ARIES is to automate the process of evaluating, potential relocation sites for units of the Army Reserve from the vantage point of military readiness.

Successful implementation of the ARIES SDSS could greatly assist the Army Reserve in its effort to ensure that a consistent, analytical approach to relocating units is adopted. Additionally it would also significantly reduce the time spent by USARC decision-makers manually tracking, calculating, and assessing the multiple criteria relevant to each facility's site desirability. The use of information technology (IT) in the form of an architecturally-driven prototype application can improve the quality, timeliness, and accuracy of the Army Reserve Unit relocation decision specifically, and of similar spatially oriented decisions in general.

B. THE MOTIVATION FOR ARIES.

The sponsor of this research is the Force Support Package (FSP) Readiness Office, a hierarchical component of USARC based in Atlanta, Georgia. This group is tasked with properly assessing and improving the military readiness of priority Troop Program Units (TPU). In the spring of 1996, several field-grade officers of the FSP approached the Naval Postgraduates School (NPS) building a decision support system to automate and expedite the selection process for relocating Army Reserve units. The manual process currently in use is unnecessarily complex, unwieldy, and limiting for USARC decision-makers, and a technical solution offers increased degrees of freedom in making these crucial readiness decisions.

Under the mantra of "*Resources to Readiness*", USARC and NPS formed a collaborative partnership to explore the concept of automated decision-making for site location decisions. The research project was undertaken by an NPS team of two academic principal investigators and three application developers. This team identified location as one of the critical factors affecting unit readiness, and based upon this assumption, implemented a computer based decision model to support the unit relocation. A spatial decision support system was used to leverage the geographic nature of the decision problem.

C. OBJECTIVES, SCOPE, AND METHODOLOGY.

1. Objectives.

The ARIES application was accomplished by integrating a multi-criteria Army Reserve unit-decision model (ARU-DM), over a dozen USARC data sources and geographic information system (GIS) into a SDSS. The SDSS was developed using rapid

prototyping based upon a component-based development heuristic, Concept-to-Code (C2C), and domain-specific architectures. The design heuristic was used as a structured means to assist the end users in conceiving the desired system, to facilitate development of a Data Migration Warehouse (DMW), and to enable the developers in systematically engineering the SDSS application to fit within the existing USARC Headquarters IT infrastructure.

The objective of this research is to describe how the C2C approach facilitated the development of a complex, map-based, multi-criteria decision support system with a small team in a matter of months.

2. Scope.

This thesis addresses only the issues involved in architecturally developing and implementing the ARIES SDSS using component-based technologies. For a more complete examination of the issues involved in framing the Army Reserve Unit Decision as a Multi-Attribute Utility Theory model, refer to LCDR Mark A. Murphy's master's thesis "*An Automated Spatial Decision Support System for the Relocation of Army Reserve Units*" (Murphy, 1997). Additionally, the development of the Data Migration Warehouse was spearheaded by LCDR Robert W. Dill and the issues of data warehousing and data quality are discussed in his master's thesis "*Data Warehousing and Data Quality for a Spatial Decision Support System*" (Dill, 1997).

3. Methodology.

The primary function of the ARIES SDSS is computation of the twenty decision parameters identified in the ARU Decision Model from the extracts of designated data sources in accordance with elicited processes and user selections. USARC specified that the ARIES SDSS should access only the existing Information Systems (IS) infrastructure

and historical databases as the primary sources for computing decision model parameters. This avoided the need for additional data collection and data normalization. The impact of this constraint on the ideal model structure, derived from user specified requirements, was carefully evaluated and implemented. Managed data resources were properly identified from those readily available to USARC. The consolidation of multiple sources into an appropriate database schema for this software application presented a significant technical programming challenge.

Due to the limited availability of programming resources and user specified requirements, the SDSS needed to integrate various commercial software tools. An RDBMS and a DSS were combined to manage and manipulate the data. Because of the spatial significance of many of the variables, a GIS was also incorporated for both displaying, selecting, and manipulating the information. The proposed software system contains five major subsystems: the ARU decision model, a single point user interface, data migration management, geographic information system (GIS), and a client-server architecture design.

To rapidly accomplish these tasks required the development team to concurrently innovate and integrate several technical disciplines in ways not previously envisioned. The final result was a SDSS prototype that incorporates the leading-edge aspects of spatial mapping, geographic selection, data warehousing, and decision support technologies.

Component-based application development based upon a systematic and architecturally-driven approach was used in developing the ARIES SDSS client-server architecture. Rapid prototyping techniques were also used to conceptualize the product design in accordance with user requirements and provide functionality for differing levels of detail in capturing the decision knowledge.

The current version of the ARIES SDSS prototype was implemented by the NPS development team using Microsoft's Visual Basic for Applications (VBA) programming language with and third-party software including MapInfo Corporation's integrated mapping components, MapInfos' Basic development language, and the Logical Decision for Windows (LDW) decision support system. The SDSS was based on the client-server computing model and used the Microsoft OLE 2.0 communication protocol as the primary means of component communication.

D. ORGANIZATION OF THESIS.

This thesis contains useful information on building applications using component-based development from a software architectural point of view. It should be particularly valuable to decision-makers, information system architects, computer programmers, and data managers. The balance of this study is structured into several modules and organized as described below:

- ***Chapter II*** - Opens with a futuristic scenario description about potential uses for next generation (NEXGEN) versions of ARIES. Additionally, a background discussion is provided about the USARC organization, their need for a SDSS and an overview of the intended system.
- ***Chapter III*** - Discusses the benefits of using component-based development in conjunction with a new software architectural paradigm (Architectonic) through the use of a design heuristic (Concept-to-Code). This chapter closes by describing the ARIES SDSS creation using the design heuristic.
- ***Chapter IV*** - Discusses the lessons learned from using component-based development and rapid prototyping when developing complex spatial applications using existing infrastructure.
- ***Chapter V*** – Provides future recommendations for further study and presents the conclusions of this thesis study.

II. BACKGROUND.

"Nothing is Built unless it is Imagined First."
NRaD "Command Center of the Future" Motto.

"Our ability to imagine complex applications will always exceed our ability to develop them." Grady Booch, *Object Solutions*, 1996.

"The hardest single part of building a software system is deciding precisely what to build." Fred Brooks, "No Silver Bullet", 1987.

A. FUTURE SCENARIO.

1158 HRS: Colonel Jackson R. McMurdo, Army liaison to the newly formed BRAC Commission 2000 on active and reserve Army unit basing matters, was looking forward to devouring his pastrami on rye when the phone rang unceremoniously. He answered impatiently and immediately recognized the frantic voice of Randolph Braid, Senator Jeremiad Beauregard's BRAC aide. McMurdo knew the drill without hearing another word. As the BRAC chairman, the senior Senator from the Southeast was taking considerable heat from the Senate Armed Services Committee about closing bases V, X, Y, Z, and corresponding Army unit relocation. An alternative plan consisting of only closing bases X, Z, and adding bases W and U had been suggested by the Committee, wherein the displaced Army units would be relocated to base Y. This was an unanticipated scenario based on previous assumptions and findings, and the Senator would need this counter-proposal analyzed before reconvening within the hour, or the whole BRAC list could lose the full support and recommendation of the Armed Services Committee.

"How does it look if we keep open bases V and Y and close bases U, W, X, and Z?" Braid asked breathlessly.

"Off the top of my head it appears to be a viable alternative, but I'll have to run it through the SDSS to be certain." McMurdo disliked these incessant fire drills from Congress and let his pique slip ever so lightly into his tone.

"Jack, we need you to disregard the condition of the equipment and facility age, and focus instead on civilian employment and impact on the community at large if those bases are closed and their units relocated."

"Consider it done. Just ensure the Senator's NetPC Pointcast is set to receive multi-broadcast material and the multimedia presenter is in automatic mode."

"Understand. Thanks Jack," said Braid contritely, hanging up his cellular phone.

1206 HRS: Turning in his chair while setting his lunch aside, Colonel McMurdo stoically started his wireless laptop and logged into the Pentagon's DoD Intranet. Accessing the U.S. Army website, he launched the "***ARGIS 21***" applet and began configuring the network user interface (NUI) for the appropriate mapping component and DSS model preferences.

He then entered the Dept of the Army Data Warehouse and chose the applicable DataMarts and databases on military leadership, readiness, competition, and status-of-forces to be integrated with the *OMB Economic/Financial* databases and *U.S. Census Bureau City/State* DataWarehouse.

1209 HRS: Selecting the "Build Scenario" function, the ***ARGIS DataMigrator*** began extracting and conditioning the desired data to a disposable database on a remote RDBMS server. Simultaneously, the ***ARGIS DataIntegrator*** began creating various communication links to the BRAC decision model, spatial mapping components, and the new ***BRAC DataDepot***.

1217 HRS: Colonel McMurdo graphically chose the Army units and bases to be evaluated and executed the scenario analysis. Using remote automation workflow technology, the process co-opted idle network processors to achieve a standard solution

in two minutes. From this baseline the Colonel reduced the weighting on facility age and condition whiling increasing civilian employment valuations. He did not like what he saw.

1221 HRS: Defining a 300 mile radius "what-if" circular analysis polygon around the target site, the Colonel redirected the ARGIS AI-Analyzer to determine which units must be moved to the target site of interest to achieve the desired outcomes. ARGIS reverse engineered the spatial environment and recommended several courses of action and budgetary schedules to achieve the objective over the course of the base closing timeline. This looked much better. McMurdo recognized the resultant solution as one minimizing the economical impact to a thriving community. He reviewed and selected the analysis summaries that respectively best supported and contradicted the Committee's argument .

1232 HRS: Using the ***ARGIS Animator***, he converted the analysis to a data visualization movie displaying the best solution with different color-coded alternatives over time as bases closed and units relocated. He graphed the economic impact thematically per site over time and marveled at the efficiency of the system.

1244 HRS: The telephone rang again. It was Braid once more. "Jack, the committee's reconvening early! What have you got? We have to tell them something."

"No problem, I have just broadcast the textual analysis to Beauregard's NetPC on the Senate floor and the ARGIS Solution should be playing on the large-screen display as we speak." McMurdo was calm and amused as he noted the contrast of his demeanor with Braid's anxiety.

"Thanks Jack," said Randolph. "You saved our bacon on this one."

1701 HRS: Jack tuned his computer into the *Inet-CNN* broadcast and allowed himself a small nod of satisfaction to see the lead story was about a remarkable breakthrough in the BRAC closure plan impasse. As newscast switched to the ARGIS

animation film, a voice-over announced in serious tones, "In a dramatic eleventh hour move, a compromise was reached in the Senate today."

As he watched the screen, his mind lamented about his previous assignment to USARC Headquarters so many years ago. Catching himself smiling, Jack spoke aloud, "Who would have thought the progeny of the ARIES project would have been so....

B. USARC AND READINESS.

The previous scenario may seem like a work of science fiction, but given the continued proliferation of information-based technologies, this future may be a lot closer than we think. Until a year ago, if the United States Army Reserve Center (USARC) needed to move a reserve unit, the analysis to manually evaluate several alternatives required a Herculean effort on the part of the Army analysts. The proposed site selection process could take up to 30-40 days to develop just a partial solution. This was due to the complexity of the decision, multiple disparate data sources, quantifying subjective judgments, and analyzing approximately 1300 possible alternatives. Today, with the successful partnership between USARC and the Naval Postgraduate School (NPS), the same analysis can be completed in minutes using a deployed stand-alone Spatial Decision Support System (SDSS) called ARIES, that shares many similar concepts to ARGIS 21, only in a more primitive and prototypical form. The ARIES concept consists of a delivery portion and a data and applications portion. The delivery system includes COTS hardware and software that provides workflow automation by calculating design model parameters. The data and application software satisfied the functional requirements for maintaining the data sources, automating business processes, and interfacing with the decision-makers.

The purpose of this research is to increase *readiness* using Information Technology (IT) in a manner that will dramatically improve the process used to select

relocation sites for Army Reserve Units. The final outcome is a repeatable and accurate process whose cycle time collapses from 40 days to an average of eight minutes per site selection scenario.

These order-of-magnitude improvements in analysis capability can be directly linked to the decentralized improvement in IT practices from host-based computing (mainframes) to a distributed, network-centric computing environment. Computing environments that previously were relatively simple have exploded into a plethora of networks, clients, servers, and component-oriented systems. Computing complexity has skyrocketed and with it the future potential for greater usage. Technological choices, although daunting, have now begun to take into consideration the issues of scalability, extensibility, and an increasingly uncertain future. Technological solutions are becoming the means to effectively and efficiently increase military readiness.

At the 1994 Software Technology Conference in Salt Lake City, Utah, the Honorable Emmett Paige, Assistant Secretary of Defense for C3I illustrated the strong interdependence that exists between information technology and military readiness in his keynote address:

> Software drives *Military Readiness*. As such, our military state of *readiness* is changing to support fighting and winning two major regional conflicts at the same time. As the world situation changes, it becomes increasingly the case that we will have to project our military capability with less notice and more precise results. The Secretary of Defense has made it clear that our highest priority must be *readiness*. We must get our software act together...and provide the support that our war fighters need, wherever they are, no matter how short a notice. And we must proceed with a sense of urgency in making our software become predictable, in the sense of its high quality and high contribution to the defense mission. (Paige, 1994)

This situation presently confronts all military commands, and the work described in this thesis takes a useful step forward in answering the challenge of increased readiness for the U. S. Army Reserve Command.

1. United States Army Reserve Command.

Public Law 101-510, the Defense Authorization Bill of November 5, 1990, provided for the establishment of a U. S. Army Reserve Command under the command of the Chief of Army Reserve (Figure 2-1). "Since the completed transition on September 30, 1992, the USARC has assumed peacetime command and control of all United States Army Reserve forces, except for Special Operations elements and United States Army Reserve forces located outside of the continental United States" (DOD IG, 1993, p. 12).

The Department of Defense's 1993 Inspector General, described the USARC mission objectives as follows:

- "Command, control, support, and ensure wartime readiness of the U.S. Army Reserve Forces.
- Organize, train, and prepare U. S. Army Reserve units for mobilization and commitment to a wartime theater of operations.
- Manage and execute all Operations and Maintenance, Army Reserve (OMAR) and Reserve Personnel, Army (RPA) funds allocated by Headquarters, Department of the Army.
- Support mobilization as directed by FORSCOM." (DOD IG, 1993, p. 14)

In order to comply with these ascribed functions, USARC "...decision-makers must be able to effectively manage the resources supporting readiness and mobilization preparedness" (DOD IG, 1996, p. 22).

> The capability to accomplish this goal is directly related to the management and availability of information. The existing...Army Reserve information systems are unable to provide timely and accurate information to decision-makers to...[enable either unit relocation or] mobilization planning...as required to meet contingency plans. (DOD IG, 1996, p. 22)

> The Army Reserve Command must often manually compile information requested by commanders and higher headquarters. Additionally updating information is time and manpower intensive and, therefore, is often not done as frequently as needed. (DOD IG, 1996, p. 23)

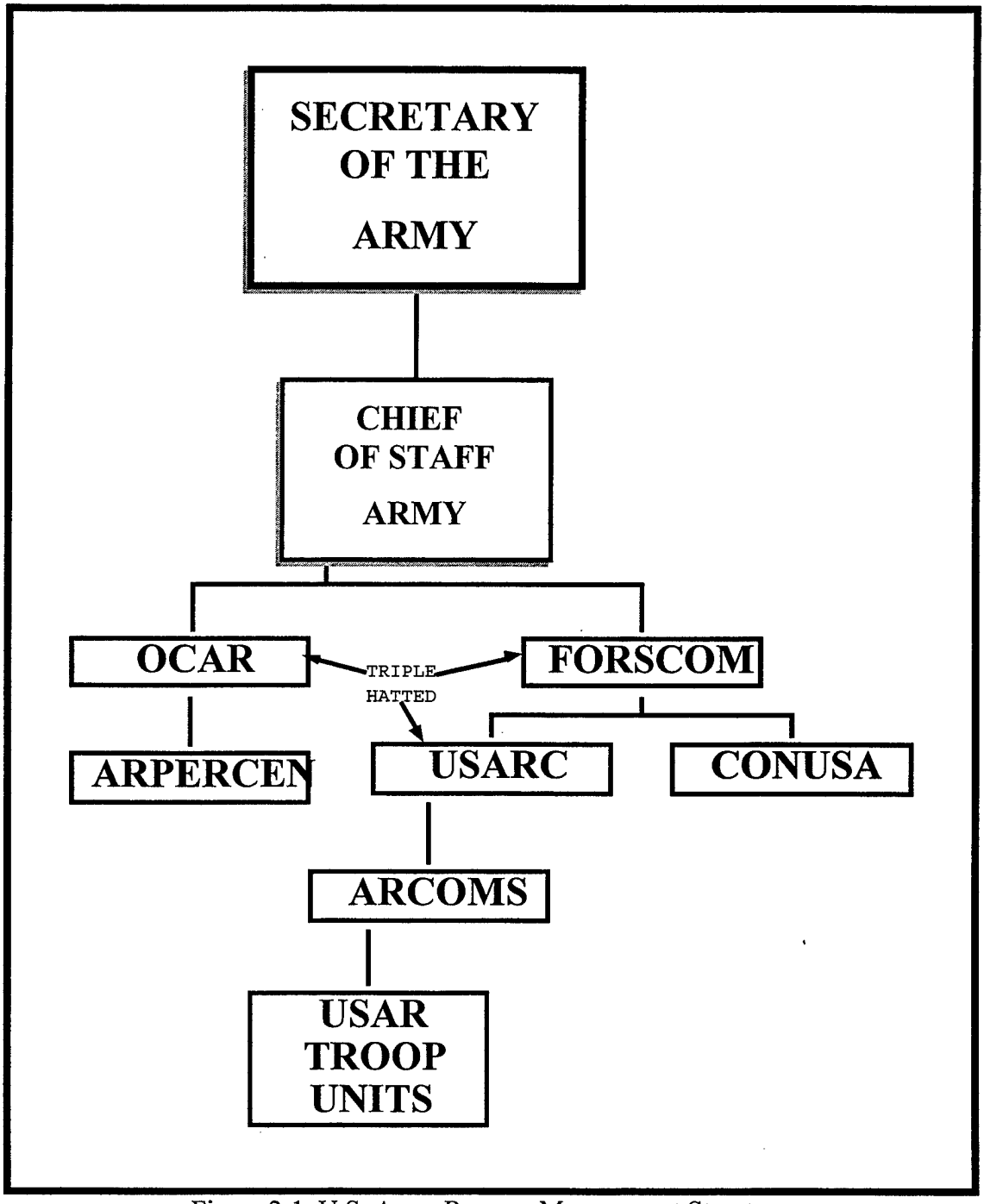


Figure 2-1. U.S. Army Reserve Management Structure.
After (DOD IG, 1993, p. 13)

With the 1994 Department of Defense stipulation to buy Commercial Off-The-Shelf (COTS) technology whenever possible, USARC is currently migrating its information systems and networks from legacy applications and databases to well-supported commercial software suites running Microsoft Windows 95 and NT-based environments.

Because of the huge expense of maintaining and operating legacy information systems, limited resources are available for new application development. Failure to think strategically about software development has lead to the formation of many stove-piped applications and "data silos of information." In order to counteract this downward spiral, the high costs of these systems must be effectively leveraged using commercial technology. Instead of deploying systems that meet USARC's short-term needs, USARC needs to develop applications capable of change and of evolving with changing business processes.

In response to this challenge, USARC established a partnership with the Naval Postgraduate School to develop an Army Reserve Unit Decision Model (ARU-DM) and SDSS prototype to automate the site relocation analysis of existing reserve facilities/units.

2. Readiness Issue.

> The sponsor of this research is the Force Support Package (FSP) Readiness Office, a recently formulated component of the U.S. Army Reserve Command. This group is tasked with assessing and improving the readiness of priority Troop Program Units (TPU) across the country.
>
> A TPU is the basic building block of the Army Reserve force, typically consisting of about 150 reservists. The TPU's that are of most concern to the Readiness Office are in the FSP, which are the units designated for rapid deployment. (Murphy, 1997, p. 1)

Military readiness can be classified into two types, *operational* and *structural*, which differ in granularity (Betts, 1995). *Operational Readiness* deals with status of

individual units whereas *Structural Readiness* deals with the overall force structure in the sense of "how soon a force of the size necessary to deal with the enemy can be available" (Betts, 1995). The focus of this research is on the operational readiness of those Army Reserve units scheduled for rapid deployment. In this context, readiness primarily refers to personnel readiness, the ability to maintain enough properly trained and qualified people.

The statistics chosen by USARC decision-makers to function as the primary indicators of unit operational readiness are Fill Level, MOS Qualification Level, and Turnover Rate. The high personnel turnover rates recently suffered by the USAR have significantly reduced readiness, With the average retraining period lasting nine to ten months, between 20 and 30 percent of the required positions are routinely filled by unqualified, non-deployable individuals (Murphy, 1997, p.13). Although Military Occupational Specialty (MOS) qualifications are not always a direct measure of warfighting capabilities, unqualified individuals directly diminish the number of soldiers available to supplement active forces.

Performance in these areas can be related to numerous location-dependent factors, ranging from access to preferred recruiting markets and distances to various training support sites. The most significant location related factor is the recruiting market; for unlike the active services, reserve units must recruit exclusively from the local population. When a unit is struggling to maintain personal readiness, sometimes the best solution is unit relocation. This thesis is based on the "...premise that, holding all other readiness variables constant, it is possible to improve the operational readiness of some Army Reserve units by relocating them to preferred areas as indicated by a variety of location-related attributes" (Murphy, 1997, p.13).

3. Army Reserve Unit Decision Model (ARU-DM).

Before the creation of a manageable model could proceed, it was necessary to clearly define the TPU relocation problem.

> Although USARC representatives suggested a variety of ways in which to understand the importance of unit location (e.g., distances, market supportability areas, overlap between units, etc.), it quickly became clear that the primary objective was to relate location to unit readiness. The decision was made to develop a [decision] model that could isolate and evaluate the location sensitive portion of the unit readiness problem. (Murphy, 1997, pp. 19-20)

> "Our methodology was to interview the USARC experts to determine:
>
> - A goals hierarchy consisting of intermediate goals and decision parameters,
> - Tradeoff functions for each parameter in the goals hierarchy, and
> - [Identify the appropriate] weights for each of the parameters." (Dolk, et. al., 1996).

Based upon the concerns and priorities of the USARC experts, the NPS development team decomposed the overall objective of *Site Desirability* into twenty measurable decision parameters. These parameters were identified as objective, measurable attributes of a desired site location and loosely grouped together in three categories: People, Places, and Things (Table I).

PEOPLE	Competition Area Drill Attendance Area Loss Rate Area Transfer Rate Average Area Manning Closing Unit Transfers IRR Available Recruit Market Reassignments
PLACES	Facility Age Facility Backlogged Maintenance Facility Condition (RED / AMBER / GREEN) Facility Operating Costs Facility Government Owned (Y/N)

	Facility Weekend Usage
THINGS	Distance to Recruiter Distance to AMSA Distance to ECS Available MOS Closing Available MOS IRR

Table I. Army Reserve Unit Decision Model (ARU-DM).

a. General Assumptions

Although this modeling effort "...does not provide a rigorous, causal model of readiness, it is assumed that the hierarchy of screening factors provides a meaningful assessment of the propensity of a given location to support the achievement of high levels of readiness" (Murphy, 1997, p.31). For the ARU decision model, the expert panel assumed that the basis for evaluation would be a single reserve unit. One alternative to this, relocating a portion or derivative of a unit, may sometimes be more appropriate, but this was not considered in the decision model. Secondly, derived reserve unit metrics had to be calculated directly from data sources available on the USARC local area network which would be implemented in the final application. Lastly, the "area of the proposed site" refers to the region within 50 miles of the zip code centroid center in which the proposed site is located.

> This proposed location will have little or no influence on where people chose to live. People will not move just to be closer to the unit or relocate when the unit does. (Murphy, 1997, p. 30)

b. Goals Hierarchy

Using the identified decision parameters, the ARU relocation decision was structured into a goals hierarchy using Multi-Attribute Utility Theory (Figure 2-2). Location data are extracted and conditioned from USARC's corporate data sources and

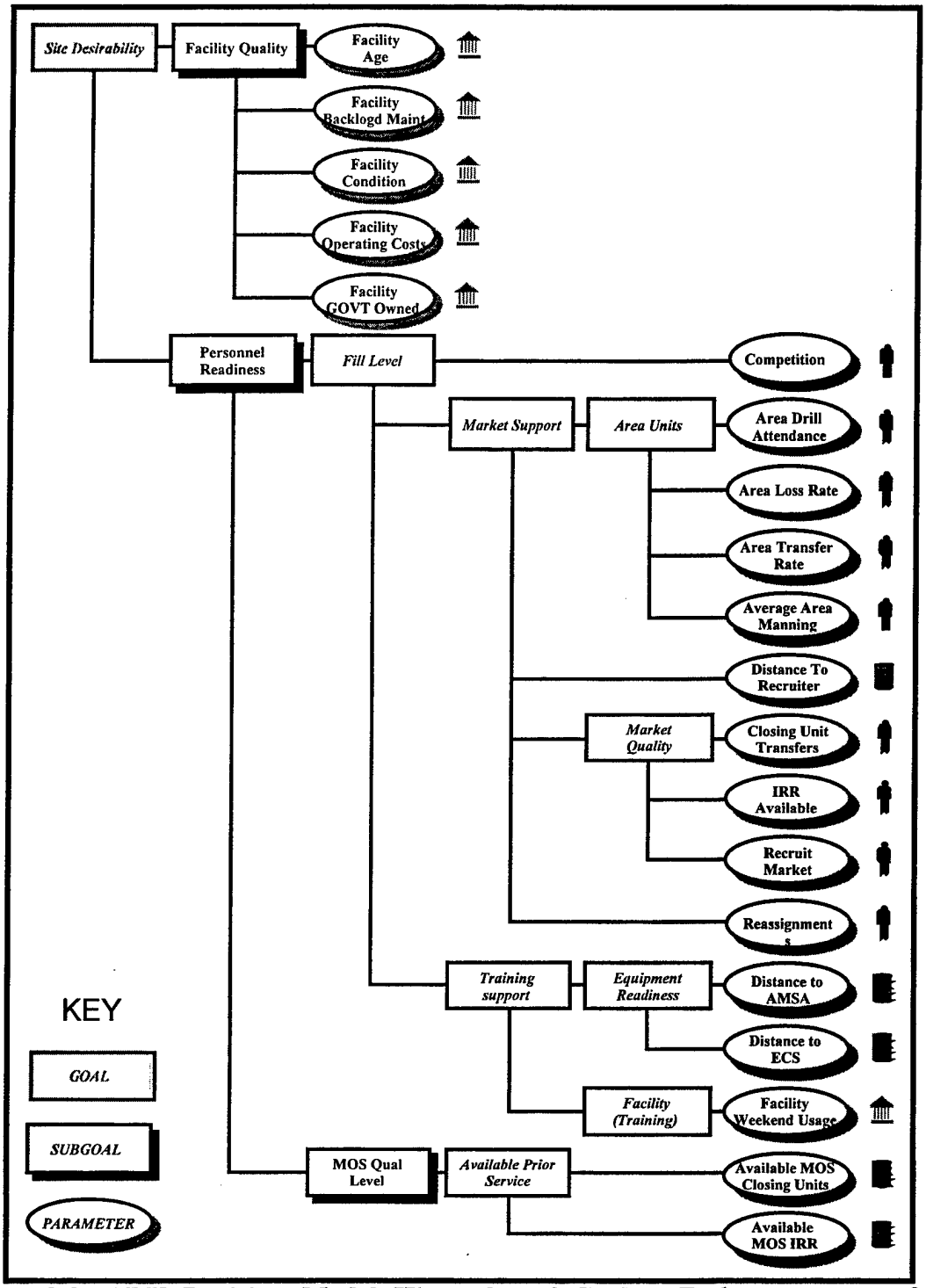


Figure 2-2. **ARU Decision Model Hierarchy of Goals.** Each component of the hierarchy is organized by its type (goal, subgoal, or parameter) and identified by the hierarchical level at which it exists. Within each section, components are listed alphabetically and levels are ordered from top to bottom. After (Murphy, 1997)

converted to common units of reference using twenty different utility function curves. There was one utility curve developed for each decision parameter. The outputs of these utility functions are repeatedly combined, based on the hierarchical structure of the ARU-DM and the relative weights elicited from the experts, to produce the overall scores used to rank the alternative sites.

c. DSS Software Component

The COTS software package selected by the NPS Development Team for implementing the ARU-DM was *Logical Decisions for Windows* (LDW). It was chosen as the DSS software component "...primarily because it supports explicit specification of tradeoff functions for each parameter as well as five different decision analysis techniques, both quantitative and qualitative" (Dolk, et. al., 1996).

4. USARC Source Data.

The SDSS prototype was originally implemented as a stand-alone Windows 95 version but is currently operating in a Windows NT environment on the USARC Headquarters' Local Area Network (LAN). One of the stipulated pretexts of the prototype was that the databases currently present on the USARC LAN are the only data sources eligible for deriving ARU-DM parameters. The primary reason for this is that USARC wanted to be able to deploy the SDSS application without having to shift the burden of administratively supporting it to the USARC IS Staff.

The native data sources are composed of a multitude of large multi-vendor databases (Table II). These databases are mostly flat file structures that were haphazardly instituted to support several independent USARC applications as they were deployed on the LAN. Some of the files are ad hoc derivatives of larger off-site mainframe-based databases, periodically updated from USAR Headquarters in Washington, D.C.

Ownership and maintenance of the local data sources is distributed amongst the primary user base of the stove-piped applications (Appendix A). Additionally, there is no centralized data dictionary or repository maintained for the data sources or the metadata that constructs them. Consequently, there is little consistency between the data source's design, structure, format, or key index fields. Database field names are different for similar data sets, and most of the files contain incomplete, inaccurate, and missing data. This complete lack of common institutional data standards greatly complicated the development of an integrated data framework for the SDSS prototype (Figure 2-3). To compensate for this unexpected lack of data quality and to provide a stable baseline database for deriving the decision parameters, the development team was compelled to

FILE NAME	*SOURCE TYPE*	*LAST UPDATE*	*FILE SIZE*	*No. RECORDS*
AMSA	PLACES	17 AUG 96	26 KB	190
CMD_PLAN	PLACES	17 OCT 96	3.29 MB	9,897
COMPLEX	THINGS	19 NOV 96	2.10 MB	1,557
ECS	PLACES	22 AUG 96	4 KB	30
FINANCE	THINGS	02 JUL 97	83.4 MB	311,793
FPS	THINGS	15 JUL 96	88 KB	1,561
FyxxLOSS	PEOPLE	05 AUG 97	85.4 MB	132,417
G17	PLACES	01 MAR 97	3.11 MB	5,869
G18CWE	PEOPLE	03 DEC 96	145.87 MB	208,416
G19TRUE	PLACES	03 DEC 96	14.35 MB	233,211
GEOREF	PLACES	02 FEB 97	213 KB	1,553
INTEREST	THINGS	13 MAY 96	4.19 MB	3,985
IRR	PEOPLE	06 FEB 97	7.53 MB	140,077
NGNON_CL	PEOPLE	09 FEB 97	636 KB	3,673
QMA	PEOPLE	14 JAN 97	2.8 MB	34,265
RPINFODT	THINGS	07 APR 97	14.33 MB	47,159
RZA	PLACES	28 JUN 96	157 KB	1,793

Table II. Source Data Tables.

concurrently design and implement a Migration Architecture System (MARS) engine, and a data migration warehouse (DMW) variant — DataDepot (Figure 2-4).

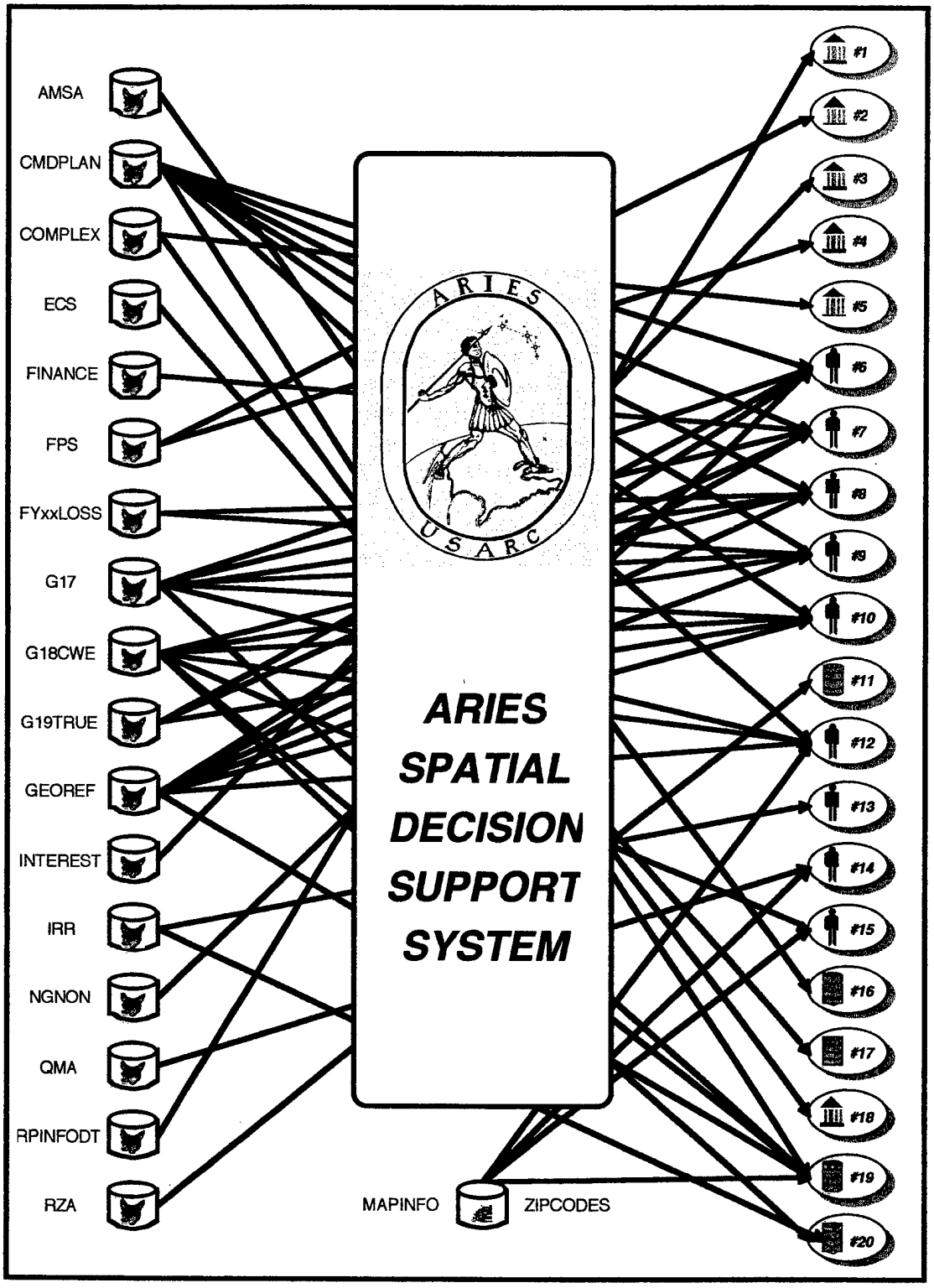


Figure 2-3. SDSS Disparate Data Sources Flow Diagram

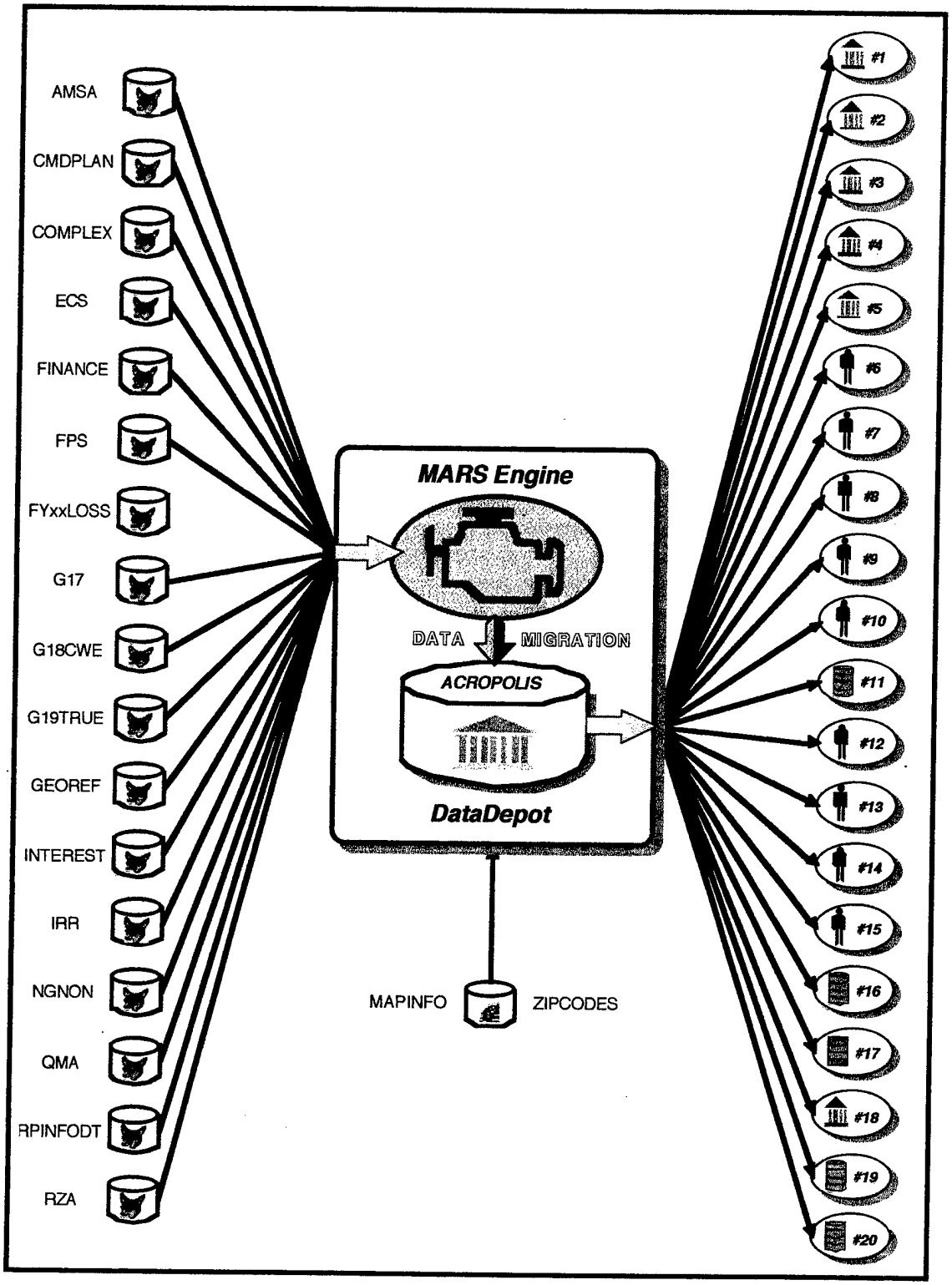


Figure 2-4. SDSS Migrated Information Flow Diagram

5. Information Systems Challenge.

Because the entire TPU relocation decision incorporates such a large number of parameters, it is nearly impossible for an unaided human decision-maker to fully consider all relevant factors. In the past these decisions were typically based upon a combination of intuition and experience, but this ad hoc process was often difficult to build a consensus for, communicate, and defend.

Frustrated with the inadequacies of the approach to such a complicated problem, the large number of characteristics per facility, the number of analyses required, and over 1300 facilities to evaluate, the Army Reserve believed it could deliver morecomprehensive relocation decisions by applying a Geographic Information System (GIS) and decision-support technology to the relocation decision process. In addition the decision support technology had to allow the USARC decision-maker to perform a variety of ad hoc analyses of multiple readiness factors to ensure that the relocation recommendation was truly warranted and was in the best interest of the moving unit. This was the inspiration for a fully automated software application that could select, calculate, and deliver spatially-oriented decision data to the ARU Decision Model. This visionary aspect of the NPS Initiative, "A Geographic Information System Approach to USAR Unit Readiness," inspired the development of just such an application — the Army Reserve Installation Evaluation System (ARIES).

In order to minimize application development time and maximize previous USARC computing investments, it was important that the automated ARIES system utilize the USARC IS infrastructure, COTS applications, and historical databases as the primary components for computing the decision model factors (Figure 2-5).

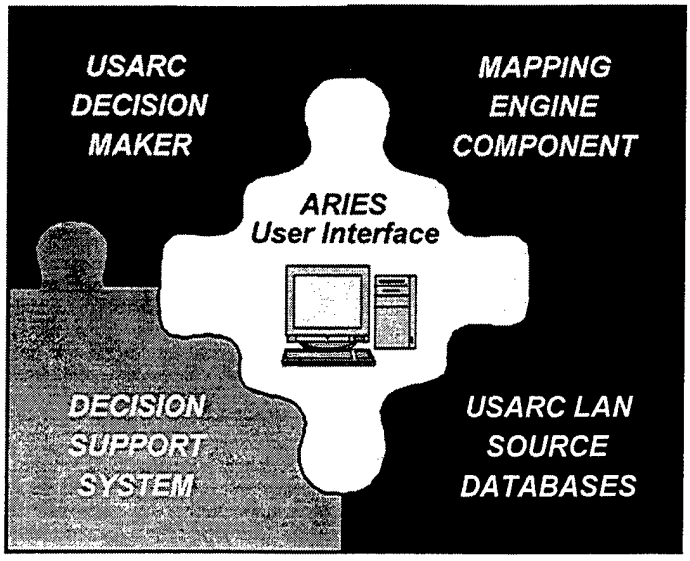


Figure 2-5. ARIES Conceptual Infrastructure Diagram.

Additionally, the NPS development team also wanted to prevent the construction and deployment of yet another requirements-driven stove-piped application (Figure 2-6), instead building one that was architecturally-driven (Figure 2-7). By this I mean an application in which the specific requirements are traceable, the architecture is not resistant to change, and the coding is kept to minimum by leveraging the existing software infrastructure. This is the real challenge of developing the USARC SDSS application.

To accomplish this requires a fundamental shift in the way developers think about and design integrated software applications. Instead of focusing on the problem as "a simple matter of programming," the development team would focus on the problem as a complex systemic concept with interdependent resources and corresponding rule-based processes. To manage this approach effectively requires a new rational design process — the process is partially proof-of-concept exploitable, fully component-oriented, and architecturally refinable.

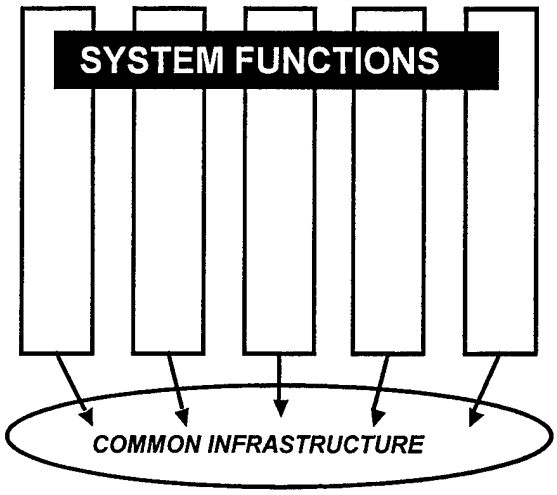


Figure 2-6. The Structure of Requirements-Driven Projects.
After (Booch, 1996, p. 15)

In the next chapter we will investigate the reasons for focusing our development efforts on architectural design and implementation, develop a component-based heuristic that incorporates architecturally-driven concepts, and specifically discuss the application development and implementation issues that went into designing and building the ARIES SDSS prototype.

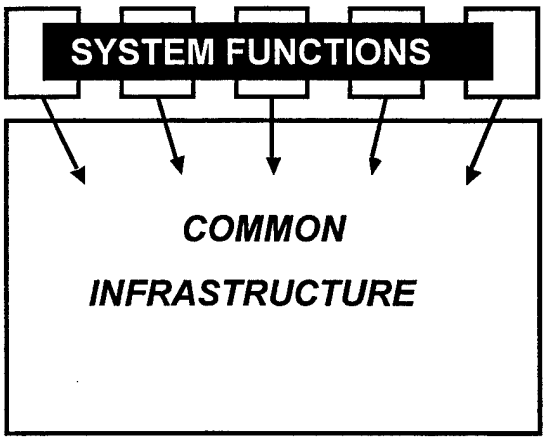


Figure 2-7. An Architecture-Driven System.
After (Booch, 1996, p. 52)

III. ARIES APPLICATION DEVELOPMENT.

"He who does not lay his foundations beforehand may by great abilities do so afterwards, although with great trouble to the architect and danger to the building." Niccolo Machiavelli, *The Prince,* 1532.

"New information technologies gradually give birth to new activities, processes, and products." John Naisbitt, *Megatrends*, 1982.

"By 1999, component software will be the dominant method of new application development (0.7 probability)" Roy Schulte, Gartner Group.

A. INTRODUCTION.

There is no single product, book, management style, or "silver bullet" that magically leads the developer or development team through a painless application development process. "But a sensible use of productive complementary tools and techniques, and a shift to a paradigm with proven advantages can over time yield the desired results" (Tkach, 1996, p. xiii). In the absence of any singular solution, the ARIES project used a "proof-of-concept" heuristic, Concept-to-Code (C2C), to guide the development team. C2C provided the structural perspective needed to manage USARC's requirements and the implementation of the decision model into a cohesive, integrated prototype using various COTS products, a fourth-generation language, and existing infrastructure.

This chapter begins with an overview of component-based development and the new computing paradigm shift it ushers forth. Next is a discussion of the generic requirements of each of the C2C phases, stages, and artifacts, and the corresponding ARIES implementation. The final major section is a summary of the ARIES implementation using the C2C heuristic process.

B. BACKGROUND.

1. Software Development: A Brief History Lesson.

Prior to the 1990's, software development consisted primarily of programmers and developers using conventional software development models (waterfall, incremental, spiral) to build applications with procedural techniques. The majority of these approaches involved the systematic use of five basic steps: analysis, design, implementation, testing, and maintenance. This was usually sufficient as long as the users' requirements were clearly defined, system architectures were understood, complexity was low, and product delivery cycle was measured in years.

> One problem with the conventional 'waterfall' software development process is that in some situations it may not be possible to define the software requirements fully at the beginning of the software development effort. Another problem is that there may be a significant risk that the design approach created to satisfy the requirements may be inadequate. A final problem is that the user only receives the software at the end of the complete and lengthy process; the user may need this capability at an earlier time. (USAF-STSC, 1992, p. 15).

Conventional software practices exploited "...the idea that if developers can get perfect specifications up front, the end result will be a perfect application" (Whitten et. al., 1994). Today this not the case. As computing power and user capabilities have increased with technological change, the following phenomena transpired:

- The application development process has become more complex today than a decade ago (Tkach, 1996, p. 17).
- "Line-at-a-time programming is...increasingly being replaced by point-and-click visual development environments" (Sarna and Febish, 1996, p. 27).
- "Interactive GUI, distributed...processing, very large integrated...databases [and] heterogeneous environments...." have become the norm. (Tkach, 1996, p. 17).

Additionally, most users have become accustomed to a window-oriented user interface (U/I), and subsequent interface expectations have risen accordingly. The application development process "...has evolved to the point that it's becoming entirely GUI-based and built up from components using techniques of rapid application development" (Sarna and Febish, 1996, p. 27). Complex GUIs are becoming standard requirements that are not easily accomplished with current techniques. The software development methods of the past are insufficient to develop rapidly the scalable, extensible, and flexible applications demanded by organizations today. Speed of delivery and completeness of stated requirements are the mantra driving the "solution du jour."

Currently, organizations want software development to take into account the hundreds of millions of dollars invested in building IT infrastructure, computing capacity, and the resultant morass of accumulated data. The "waterfall" models have failed to provide the "silver bullet" solution to application development in today's "Buy-It versus Build-It" mentality. In turn, they have become the legacy methodologies of the same legacy data, applications, and systems they helped to create. A better solution needs to be found. One answer lies in the increasing popularity of component-oriented computing; with the continued maturation of the software industry, component-based development is now seen "as an important step toward the industrialization of software that can help to transform programming from an arcane craft to a systematic...process" (Taylor, 1991, p. iii).

2. Component-Based Revolution

Component-based development, the ability to design and build from definable objects without the benefits of inheritance or polymorphism, is ready to be the next great advance in software development. Although the concept of constructing applications with reusable code as been available since the object-oriented revolution, it is only just

recently that critical mass has occurred in the computing industry. The pieces are in place for component-based application development to achieve mainstream status amongst today's IT organizations. (Chappell, 1997, p. 1)

The major reasons for this are:

- The growing use and acceptance of Microsoft's Component Object Model (COM) as the dominant standard for PC-based component computing.
- Components are moving off the desktop and beginning to play an important role in the creation of client/server applications.
- Broad support for designing and developing components and component-based applications with third-party tools and languages. (Chappell, 1997, p. 3)

> As a developmental process, component-based development (CBD) "...will supplant earlier programming archetypes, such as structured programming and object-oriented programming, as the approach most likely to yield significant productivity and reusability benefits. Its emphasis on iteration sounds the death knell for the traditional 'waterfall' model of the entire development process. One of the reasons that CBD will gain broad acceptance is that it offers ways to address the complete range of software challenges, from operating system services to client/server development." (Spitzer, 1997).

Component-based development is not without its own baggage, however. As more and more organizations and vendors shift their focus to take advantage of this new wave in computing, the more susceptible they will become to the unintended consequences of the "revolution." One of the biggest challenges in the component-oriented "revolution" so far is the utter lack of any kind of established methodology for effectively guiding the conceptualization, analysis, and implementation of a user-centric application, or managing component-based construction from infrastructure components.

Additionally, there is not a widely accepted standards-based component architecture from which to develop applications. Compounding this problem is the fact that almost every component-oriented solution is unique in its configuration and implementation. Traditional software development methods do not apply. The model

currently most applicable is the one an organization develops for its own IS operations and IT environment (Hamilton, 1994, p. 43).

The other reality is that open component architectures — with their interdependent servers, databases, clients, and protocols — have a multitude of potential connection point failures" (Hamilton, 1994, p. 43). Although, we are still better served by past software development practices, it continues to remain horribly expensive to construct custom software for automating a particular organization's unique business processes (Udell, 1994, p. 47). "...Today's major problems with software [development] are not technical problems, but management problems" (DOD-USD, 1987). Developers need an efficient means in which to manage and "pre-engineer" modular applications, databases, and information flows prior to deployment in a real world environment. Because the systematic construction of complex software applications from existing components remains an ever elusive goal (Garlan et. al., 1995, p. 17).

From a technical perspective, a combination of component-based development and the latest advancements in software architecture such the ARIES SDSS is required. The prerequisites for this would need to include:

- A widely accepted component-oriented infrastructure.
- Standard Domain-Specific Software Architectures (DSSA) to guide development and enable rapid assembly.
- Commercial and technical environments that fully support CBD integration.

This alone may be the most important paradigm shift in software development in decades. The dawn of the "industrial revolution of software," whereby software products are manufactured in accordance with a specified structure and process, is upon us.

3. Architectonic Paradigm: The Shift to Structure

"Today the term [paradigm] is widely used to define a broad model, a framework, a way of thinking, or a scheme for understanding reality" (Tapscott, 1993, p. xii). The Architectonic Paradigm — of or pertaining to principles of architecture — mandates a shift in conventional thinking about how developers build applications. It requires an architectural approach that spans both the business and technical arenas, and that adopts the perspective of software construction not as a craft but as a discipline (Davis, 1995).

This new paradigm consists appropriately of the following aspects:

- Simplicity - easy to comprehend and communicate.
- Task-centric - must map naturally to the customers' world.
- Changeable - enable rapid, flexible, and scaleable paths.
- Manageable - support heterogeneous platforms, applications and architectures.
- Leverageable - migrate legacy systems and support COTS integration.

In short, the Architectonic Paradigm is one in which the overall focus shifts from requirements-driven to architectural-driven development. It "has all the benefits of requirements-driven style, as well as the favorable characteristic of encouraging the creation of resilient frameworks that can withstand shifting requirements and technological calamity" (Booch, 1996, p. 21). But the days of designing an information system in a vacuum are over. The problems of today are up to 50 times larger in complexity than the problems of yesteryear, and the difficulty of developing "good enough" solutions is growing at an exponential rate. The developer is always trying to fit a new piece into a puzzle that has already been framed and to deal with quirky methods of interfacing with the database(s). A developer can no longer afford to build software applications from scratch without duly considering the existing infrastructure and several components — which are themselves architectures (information, application, and

technical) — contained within the structure. The Infrastructure investment holds a central position in any planned and future application development (Figure 3-1).

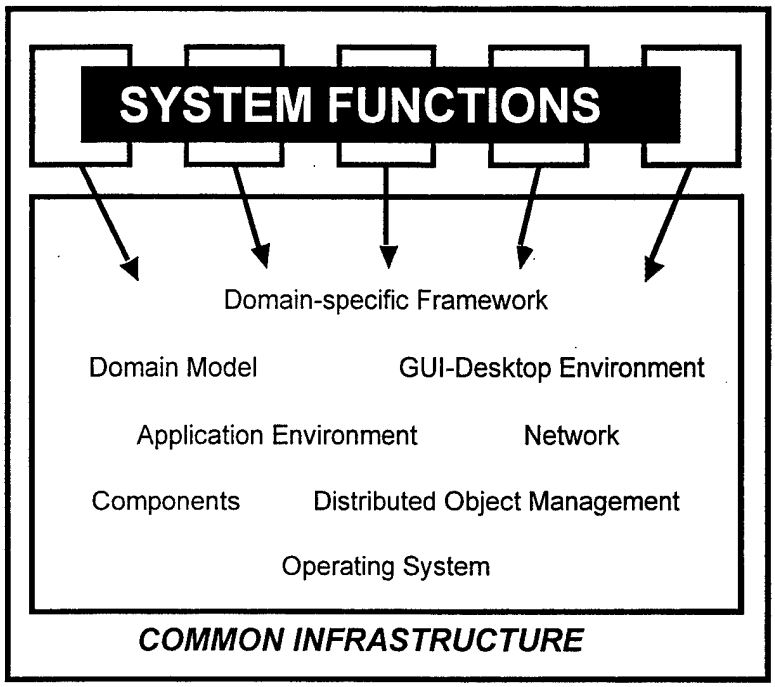


Figure 3-1. An Architecture-Driven System. After (Booch, 1996, p. 52).

4. Infrastructure: The Invisible Hand of Structure

The previous chapter briefly introduced the notion of architecture-driven programming and illustrated how it relates to infrastructure. The central idea is to encourage the minimal programming of new applications by using software architectures and COTS products already present within the desired operating environment. An example in a Windows-based environment would be to develop an application's U/I using Window's API routines rather than creating new objects and classes in the C++ language. Additionally the use of validated infrastructure components require that new applications only need to be tested at the points of interaction between existing components and thoroughly tested for any custom designed components. This allows the

developer to safely assemble and application as opposed to building it. The guiding principle is "*Buy the Basics, Custom Build only what provides Value*" (Knowles, 1997).

Many mature organizations, already dependent on information systems, are struggling to implement this "Best-in-Breed" approach while still maintaining their existing lines of business applications. To accomplish this, organizations must effectively forge a vision for an enterprise architectural foundation with a (capital "A") and develop domain-specific frameworks that support the integration of single application architectures (with a lower-case "a"). Perhaps, the best approach is to fundamentally leverage the existing infrastructure in a manner that permits organizations and developers to manage heterogeneous components, business rules, and legacy data from an architectural perspective.

The ARIES development team devised the Baseline Infrastructure Archetype (BIA) Model to accomplish this objective, and to better understand and analyze the infrastructure. Although the BIA Model is generic, it forms the centerpiece of the Architectonic Paradigm, and offers guidance for evaluating the features of available components and tools used to solve problems relevant to the unique circumstances of individual applications. Essentially all of the pertinent technology fits into the multi-layer triangular model (Figure 3-2) that incorporates all three infrastructure subarchitectures: information, application, technical. These subarchitectures provide a triad of interaction for all systems within the Architectonic Paradigm:

- ***Information Architecture*** - describes the content, behavior, and interaction of all the business and information requests from applications and technical components. The information architecture provides the database objects and services for application development and provides a information framework for other requesting resources.

- ***Application Architecture*** - defines the fundamental services and business rules within the applications domain. This architecture transparently and seamless supports the logic and data processing of the user interface service requests. The components of the information architecture are modeled,

designed, and implemented in terms of the business rules and processes developed in the application architecture.

- ***Technical Architecture*** - specifies the actual technologies (components) and the software tools that supply the requested services to and from the application and information architectures. It is used to implement the applications. (Aronica and Rimel, 1996)

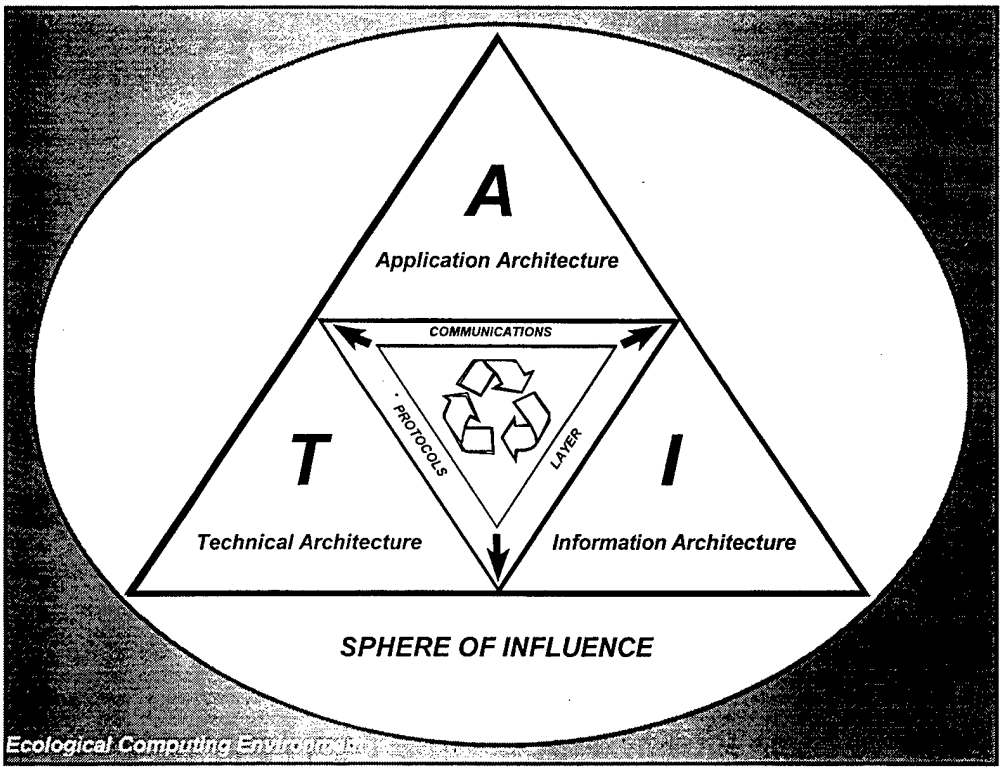


Figure 3-2. Baseline Infrastructure Archetype (BIA) Model.

The logical layers of the BIA Model that function from the epicenter (interface layer) outward are:

- ***Interface Layer*** - Provides client-side services to the resources and user. For a particular user the layer enables access and interaction with application resources.

- ***Communications Layer*** -Functions as middleware to connect and manage the interface clients with various requested resources.

- ***Resources Layer*** - Consists of the DBMS, databases, and data; the applications, parts of applications and components; technologies and tools that implement the applications. (Hurwitz, 1997)

Resource interactions that do not require client involvement occur at the angular margins, and the byproducts of interaction are conceptually maintained within the circular segments of the "sphere of influence". More simply, the sphere represents the scalable world in which the information system functions. Whether it is a standalone desktop computer or a mainframe system, the baseline structure is physically different but logically similar. The ecological environment in which the spherical microworlds operate provide the common platform services:

- Base Operating System (e.g., Windows NT).
- Networking Communications (e.g., NetBUI, TCP/IP, etc.).

> The layers in this model are not the same as those found in a typical three-tier client/server model. There is no single service with a single piece of middleware, no single graphical user interface, and no single type of data all tightly connected. You should think of each layer as a category -- an organizing principle. (Hurwitz, 1997).

This layer approach allows developers to plan the function of various pieces of the architecture and determine what to work on first. A specific example of this model will be provided later in the next section.

The next logical question is "How can the benefits of architecture be effectively extended to component-based computing?" The ARIES development team's response was to devise and institute a methodological heuristic, *Concept-to-Code*, that embraced the Architectonic Paradigm and provided a structural perspective and the means to formulate a cohesively integrated application. This will be the focus of our discussion in the next section.

C. CONCEPT-TO-CODE HEURISTIC

1. Overview

The main disadvantage of major information system development, aside from architectural and infrastructural issues, is that it takes too long to implement full production products from untested concepts. It is very difficult to get beyond theoretical constructs to practical implementations.

What is needed is a new kind of application development, one supported by a set of design ideas that can accommodate pre-existing applications, software infrastructure, and COTS products, and one employing a systematic approach independent of tool, style, or technique that acts to guide and direct the collaborative efforts of the users and developers, and that incorporates Grady Booch's five elements of a successful project application:

- Focus ruthlessly on the development of a system that provides a well-understood collection of essential minimal characteristics.
- The existence of a culture that is centered on results, encourages communication, and is not afraid to fail.
- The effective use of object-oriented analysis and design.
- The existence of a strong architectural vision and principles.
- The application of a well-managed incremental and iterative development life cycle. (Booch, 1996, p. 25)

Concept-to-Code (C2C) is just that: a way of getting from a concept envisioned by a organization to the specific computer code needed to achieve it. C2C's primary cognitive focus is to prevent the production of ill-defined software early in the conceptualization process. The heuristic achieves this by exploring and implementing practical solutions in a manner that is independent of a particular computing language,

DBMS, and operating environment. Simply stated, by using rapid application prototyping techniques and the participatory involvement of the intended users, an organization can get exactly what it needs in less time than by conventional methods.

"In business system development, satisfying the needs of the users is particularly challenging because even the user cannot know what he wants and needs until he sees the system in action" (Scharer, 1986, p. 59). Because written specifications are not a good way to communicate with the user, rapid application prototyping was selected by the ARIES development team as a practical approach for developing the U/I and the application. Prototyping in this case becomes a method of system construction as well as a technique for system definition (Scharer, 1986, p. 60). Additionally, it is critical to remember the following when using rapid application development (RAD):

- RAD only works when you understand the business problem.
- RAD should not be tool-driven or tool-dependent.
- RAD is only effective when it's part of a sound application development process. (Linthicum, 1997)

Achieving this RAD effort requires close communication and collaboration between intended users and a small, highly skilled development team (e.g., three to eight people).

> The task is to develop a technical artifact for a client or user with more or less clear and stable requirements. To cope effectively with the uncertainty of this task, an experimental approach is taken in which various models, prototypes, and versions are tried to reach a satisfactory solution. (Dahlbom, 1997, p. 84)

By using a methodology or process that integrates the combination of controlled development and rapid prototyping, a developer can ensure minimal risk and maximum benefit in design.

In order to effectively organize the RAD process, C2C was visually conceived as a three-layered closed-loop pyramid structure flowing progressively from top to bottom and iteratively throughout the structure (Figure 3-3). Each layer represents a distinct

"phase change" of the collaborative process, or the completion of an iterative milestone of the developing application. Each phase is further sectionalized into two smaller parts called "stages." As indicated, each stage flows into each other and acts to facilitate the focus of the user and developer. The resultant IN/OUT processes of each stage are byproducts of this interaction.

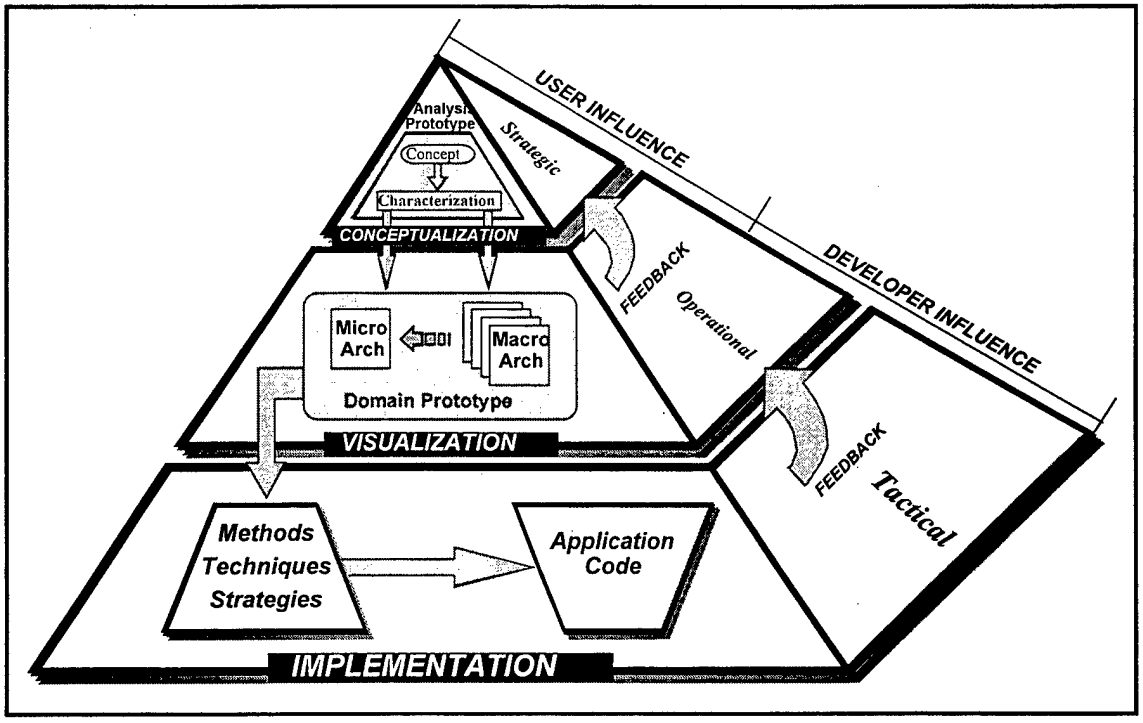


Figure 3-3. Concept-to-Code (C2C) Heuristic Diagram

The *Conceptualization Phase* is focused on identifying and clarifying the concept or problem to the extent that the prospective users are able to articulate and the developers are able to assimilate. The artifacts of the conceptualization phase are utilized by the *Visualization Phase* to more thoroughly analyze the problem domain, define and design stable component architectures, and address the riskier elements of the project. This phase will more than likely require several iterative feedback loops to properly create the U/I prototype and application architecture in accordance with the user's

requirements. The *Implementation Phase* uses the approved byproducts of the Visualization Phase employing appropriate application development principles and fourth generation language coding skills of the developers.

Prior to the transition from each phase/stage, a formal review and informal walkthrough should be conducted to ensure the resultant artifacts are satisfactory and design decisions sound. Usually peer-to-peer walkthroughs prevent a perpetuation of inconsistency and incompatibility between user requirements and developer objectives. "Formal reviews are intended for acquisition management users and interested parties to conclude a software development phase by approving the products and other results of that phase" (USAF-STSC, 1992, p. 55).

A completed cycle of the C2C pyramid and acceptance of the application by the user represents a final incremental version that is ready to be deployed in the desired operating environment. "The process is incremental in the sense that each pass through the...[C2C] cycle leads a project to gradually refine its strategic and tactical decisions, ultimately converging upon a solution that meets the end user's real (and usually unstated) requirements, and yet is simple, reliable, and adaptable" (Booch, 1996, p. 29). Later in this chapter, I will specifically address each phase and stage in turn, but for now it is easier to view each phase and stage as a transition point.

The apex of the pyramid is the highest point of general abstraction and becomes more specialized as the developer gravitates to the foundation. Only as the user-developer collaborative ensemble descends the structure do specific commitments to particular languages, standards, and operating environments emerge, all the while providing a mechanism for feedback that enables architectural refinement and iterative development of the concept. The bisection of the upper half (user-driven) and lower half (developer-driven) of the pyramid, indicates the primary focus of the collaborative effort.

The shaded boxed regions in the first two phases illustrate where rapid prototyping occurs dependent on user requirements:

> The ***Analysis Prototype*** is an aid for exploring the problem domain. It is meant to capture users' input and show proof-of-concept. The analysis prototype is not meant to be used as a basis for development and it should be discarded when it has served its purpose. The final product should use the concepts exposed by the prototype, not its code.
>
> The ***Domain Prototype*** is an aid for the incremental implementation of the solution. It can be used as a tool for staged delivery of subsystems to users and other developers. It demonstrates the feasibility and viability of the implementation and will eventually evolve into a deliverable product. (Tkach, 1996, pp. 52-53)

Therefore the amount of prototyping is directly proportional to the degree the user is able to effectively articulate what s/he wants, needs, and desires (Figure 3-4). This dependency upon the judgmental influence of the user diminishes from phase to phase (Figure 3-5). Because of this dependency the prototyping for the ARIES SDSS will be discussed separately at the end of each phase narrative — each prototype situation is uniquely different from the previous, and at a minimum, the U/I-centric nature of today's applications, increasingly demands prototyping to achieve the best fit, form and functionality in a very short time.

This coordinated prototyping capability is what makes C2C far different from typical application development. It lends itself easily to rapid application development yet is iterative at each phase. "Without a [process] for managing change, all but the smallest, simplest projects are at risk of wandering out of control and adding time to the development schedule" (Forte, 1997, p. 121). The heuristic on the whole is not a cookie-cutter technique, but takes a "systems view" of the concept to be tested and develops the application in steps from concept to system to subsystem.

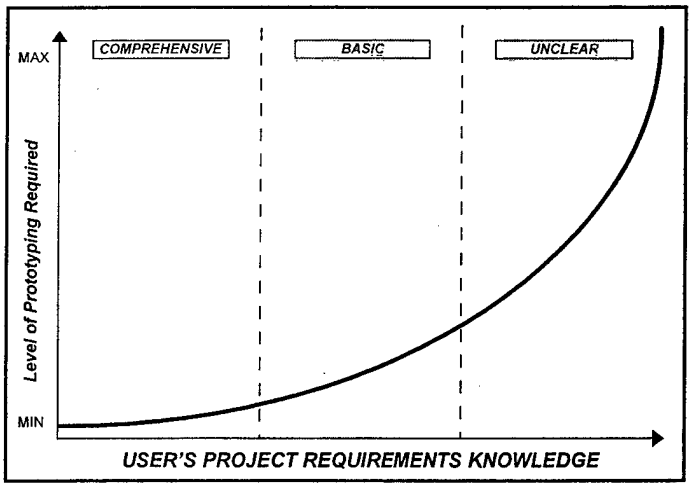


Figure 3-4. Level of Prototyping Required Graph.

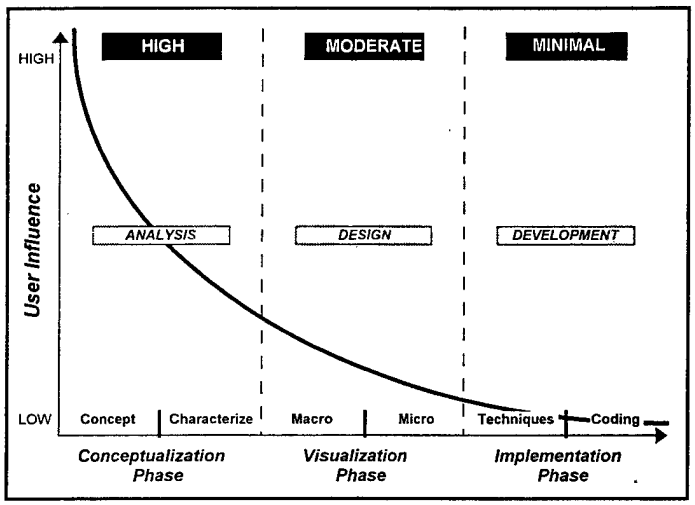


Figure 3-5. User Influence on Application Development.

With this in mind, a more complete definition of C2C is the following:

C2C is a formalized iterative construct, a heuristic, for developing "proof-of-concept" applications in a top-down, component-oriented architecture in which users' objectives, business rules, and organizational data are defined and related to one another at the conceptual, technical, and implementation levels.

The tenet principles upon which this heuristic was based are as follows:

- Manage the complexity of small to medium sized applications (< 100,000 LOC) through the use of component-based techniques while still embracing legacy systems.
- Focus development and modeling efforts on components whose function is determined heavily by the user interface and interactions with relational databases.
- Where applicable, adapt to changing requirements by maintaining a strict adherence to open IT standards-based architectures to ensure *interoperability*, *portability*, *scalability,* and *data exchange* (Libicki, 1994, p.3).
- Ensure application efforts are user-driven, not technology-driven, by leveraging the user and developers intellectual capital through rapid application prototyping.
- Conduct integrated and continuous testing with the C2C process while maintaining a "high level of stability throughout the project by constantly evaluating the architecture, interfaces, and...by preventing the accumulation of latent defects" (Forte, 1997, p. 122).

Now that the basics and underlying principles of C2C have been explained, we can delve into the specific regions of the heuristic and how they help to direct the development of the ARIES application.

2. Conceptualization Phase

Concept-based exploration begins with a broad concept — a "term...meant to suggest a framework that is found useful in organizing ideas and suggesting actions" — and proceeds into a characterization plan to determine and understand the processes of the organization, and identify the top level requirements and feasibility of the concept (Scott-Morton, 1984). It is essential to understand the business-centric aspects of the organization prior to developing the concept. The main goal in the first step of this phase is to identify the concept to be proven, and characterize it in terms with which the user is familiar in preparation for later software development.

The second step in the conceptualization process is identifying the specific capabilities required to execute the concept by focusing on defining as much as possible the elements that best characterize the organization's stated concept. The goal is to develop a characterization of what the application will accomplish without getting into the details of how the system actually functions. The term characterization is used because process analysis and data gathering are not exhaustive. It is not necessary to document every detail. Only enough detail is required to allow informed decisions to be made in the creation of the component architectures and U/I prototype. Without the insight that a baseline characterization provides, there is a risk of devising a system architecture that is difficult to improve. If the customer already possesses much of the baseline data, more time can be spent on characterizing the environment. Iterative development will subsequently refine the characterization to a specific system.

Much of this information is gathered through a process of interviews with all the individuals (Management, IT staff, and end users) with a vested interest in the application; as well as a thorough audit of the existing environment. It is important that the intended users of the system be fully involved and participative in the conceptual design process. Conceptual errors made in the design are laboriously difficult to correct once coding has begun. Developers prefer to carefully analyze and evaluate the conceptual design before proceeding to the Visualization Phase. If the user has difficulty articulating or characterizing the concept, it is the responsibility of the development team to utilize prototyping as a means to elicit the desired information. The Analysis Prototype can help to "minimize development risks due to incomplete requirements, and to assess whether a user interface can be developed that will allow the designated system to operate effectively" (Dolan, 1994).

Some of the basic concept characterization elements to consider are (not inclusive):

- Top-Level Requirements.
- Scope.
- Timeline for Product Delivery.
- Performance Criteria.
- Technology Base and Infrastructure.
- Business Rules, Processes and Logic.

The user is not asked to freeze the functional specifications during this because specifications can be revised or changed even after the analysis prototype refinement is in progress. Accordingly, this will require a different mind-set than previously conducted by software development teams in the past.

a. ARIES Concept Stage

The concept USARC was interested in investigating was whether or not the manual process of relocating Army Reserve units could be automated using existing data sources and computing infrastructure. Specifically, could a Geographic Information System (GIS) be integrated with an ARU-Decision Model to provide users with the ability to graphically select and compare reserve units through a common user interface. The resultant solution, a Spatial Decision Support System (SDSS), would be a new capability that allows the decision-maker to "pull" only the spatially pertinent data about desired facilities and "push" the results into the DSS model.

b. ARIES Characterization Stage

The desired ARIES system could best be characterized as a software application that spatially selects, derives, calculates, and imports the user-identified decision parameters into the ARU-DM using USARC data sources.

(1) Top Level Requirements

- SDSS application must be a standalone "proof-of-concept" prototype that will execute on a USARC-furnished laptop computer running the Microsoft Windows 95 operating system.
- Allow the decision-maker to manage the selection of the legacy data sources on the USARC LAN.
- Implement a single, consistent user interface that allows the decision-maker to graphically or manually select the facilities to be compared.
- Automate the calculation of ARU-DM parameters, using USARC business rules, pre-determined data processes, and a GIS.
- Enable the batch printing of pre-determined reports from a default preference set.
- Allow the decision-maker to change parameter weightings and conduct "what-if" analysis on the completed scenario.
- Use as many USARC infrastructure components as possible without adding additional administrative burden to the IT staff.

(2) ARIES Scope

- *Data Sources* - U.S. Army Reserve Databases (National); Approximately one gigabyte. The functionality of the SDSS had to be proven with USARCs' existing systems, using data the decision-makers were familiar with.
- *Inputs* - Unit Identification Code (UIC) of the moving unit and up to four Facility Identification (FacID) unit alternatives.
- *Complexity* - Support no more than 10 USARC users at a time in the LAN environment.
- *Constraints* - For calculating applicable decision parameters, spatially select the closest AMSA, equipment and recruiting sites; Conduct all spatial selections within a 50 mile radius of the proposed site; Use USARC's command plan as a master list to validate UIC's and G17 files as the master list to validate FacID's; Indicate inaccurate or missing data for decision parameters with code (**-999**).

(3) Timeline

- Develop standalone prototype in approximately six months.

(4) ARIES Performance Criteria

- Conduct each evaluation scenario (up to four alternatives) within two hours maximum; Verify each user input using master validation lists; Batch automate as much as possible in order to minimize user interaction and training.

(5) USARC Technology Base and Infrastructure

- *Hardware (standalone)* - Dell Latitude laptop computers with 90 MHz Pentium processors.
- *Hardware (network)* - Numerous multi-vendor desktop computers with Pentium Processors.
- *Operating Systems* - USARC LAN running Microsoft Windows for Workgroups 3.1 with a plan to upgrade to Microsoft Windows NT 4.0 in Fiscal Year 1997. USARC laptops utilize Windows 95.
- *COTS* - MapInfo Professional and MapBasic 4.0 mapping software package. Microsoft Office 95 Suite.
- *4GL* - Microsoft Visual Basic 4.0 Professional Edition.

(6) ARIES Business Rules and Processes -- Appendix B

c. *ARIES Analysis Prototype*

Because users cannot always articulate what their business requirements are or their preferences for the user interface, two forms of analysis prototyping were used to better conceptualize the ARIES system.

(1) Contextual. Because direct interaction between USARC and NPS was restricted to a maximum of two days per month a contextual style (story-boarding) of prototyping was used to elicit from the users the desired functionality of the common user interface in terms of visual appearance, menu choices, program controls and toolbar capability. The idea was to distill the "signal from the noise" by starting from the user's perspective and rapidly discussing, diagramming, and receiving feedback about various interface ideas and alternatives. As each iterative cycle was completed, the

interface became more refined, until all sides agreed upon a interim design. The final computer-based design was completed later as a byproduct of the component architecture prototype in the Visualization Phase.

Because the application was to execute in a Microsoft Windows-based environment, it was important to the users that the user interface mimic and leverage Windows-based functions as much as possible. The main objective was an application that required a minimum amount of training for experienced Windows users and fulfilled all of the stated requirements. To achieve this, the collaborative team agreed that all program views and controls would be subordinate to, and represent only one item at a time within the confines of a master window view. This was needed to maximize user comprehension and minimize confusion. The particular views and controls are as follows:

- Provide a Map View that enables the user to graphically select desired units and monitor the progress of facility comparisons that are spatially selected and calculated.
- Provide a Selection Panel that enables the user to manually input Moving UIC/FacIDs and monitor the status of facility comparison calculations.
- Implement button controls for accepting, starting, resetting and exiting the application.
- Implement a menubar and toolbar that allows the user to manipulate the COTS products in accordance with user requirements.
- Provide feedback to user in the form of a statusbar and a selected facility information view.

(2) RAD-based. A second analysis prototype used a combination of 4GLs, Delphi 2.0, Visual Basic and MapBasic, to develop a rapid application to audit and validate the USARC business rules and data logic for deriving the ARU-DM parameters. This was accomplished by creating relational database links to a subset of the data sources — Army Reserve Units in Pennsylvania — in their existing

format, computing the decision parameters in accordance with the business rules, and outputting the results to a common data file format (e.g., Tab delimited). Additionally, a custom component was created and developed using the MapBasic spatial application development language; a programming language associated with interacting and automating the MapInfo application. Essentially, the MapBasic component functioned as a workflow script that spatially selected the data necessary to calculate several decision parameters within a 50 mile radius. This proved to the collaborative team that the most critical aspect of the prototyping process, using USARC LAN files without modification and coordinating various heterogeneous COTS products, could be controlled by a common user interface. By prototyping ARIES in this way, we were able to obtain clear guidance on how to meet and comply with USARC's unique military requirements and business rules.

d. ARIES Conceptualization Artifacts

Before proceeding to the next phase, a formal walkthrough review was conducted with all involved to ensure all user requirements had been gathered and the concept properly characterized. It was important to avoid "gold plating" the requirements (e.g., providing more functionality and features then the user needs) by realistically examining the application's proposed concept and the organizations existing IS/IT infrastructure. The resultant ARIES artifacts of this phase were determined to be the following and are detailed in Appendix C:

- Conceptual Overview Diagram
- ARIES Process-Flow Diagram.
- USARC Business Rules Specifications (Appendix B).
- User Interface Interim Layout.
- User Interface Final Layout

3. Visualization Phase

Once the business issues are characterized and the application requirements have been defined, the next step is to turn these abstract requirements into an actual tangible architecture. The Visualization Phase does this by providing the means to synthesize the user requirements into functional architectures and component interactions. Unlike current software development practices, C2C does not attempt to clarify the difference between requirements and design. The goal is to facilitate the construction of the needed application using the most appropriate architectures.

The C2C architecture design involves multiple levels of abstraction. At one extreme is the reference or domain-specific software architecture (DSSA); the other is the application architecture itself. The. DSSA represents the most general and abstracted form whereas the application architecture represents a specific design solution. In this phase, we categorize the two architectures as macro and micro. The macro-architecture illustrates the "what" and the micro-architecture illustrates the "how" of the application being designed. While the micro-architecture represents a specific technical solution and will change as the application is revised, the functional macro-architecture will remain the stable domain structure. Based on the requirements identified in the previous phase, the macro-architecture is partitioned into modules and the function(s) of each module are defined. The micro-architecture then entails the identification and selection of component software to fulfill theses functions. More simply, the micro-architecture key concerns are functionality and performance; whereas the macro level concerns focus upon the management of complexity.

One of the problems of identifying components is that future users of the application often view all of their processes as unique, and therefore desire custom-design components to handle this uniqueness. This is where a proper balance of "Buy -versus-

Build" strategy is needed by application developers; because some of the components may be available in the form of COTS products, and some components may have to be custom developed for the specific function they are required to implement. A good rule of thumb is "Build Business Value, Buy the Basics" (Knowles, 1997). Secondly, a component should be linked to an accepted standard and it must earn its way into the application architecture. The use of architectural styles such as these have several significant benefits:

- Promotes design reuse within the problem domain.
- Often leads to significant code reuse and shared implementations.
- Enables better comprehension of the system's organization if conventional structures are used.
- Use of standardized styles supports interoperability.
- Permits specialized analysis by constraining the design space. (Monroe, 1997, p. 45).

We envision multiple passes through the Visualization Phase with the macro-architecture being solidified first, the U/I in subsequent passes, and finally the component architecture (micro). Refinement occurs at each iteration with the end user closely engaged in the resultant architecture.

a. ARIES Macro-Architecture Stage

Because SDSS was an innovative concept, there was no pre-existing reference architecture/DSSA upon which to base our design. With this in mind, the collaborative team utilized the artifacts from the previous phase and the BIA model to design a macro-architecture (Figure 3-6):

- *Point of Access (POA) Module* - the prototyped U/I
- *Application Module* - includes the ARIES Business rules and processes to capture the ARU-DM decision parameters and transfer them to the DSS.

- *Technical Module(s)* - includes the capabilities to display maps, spatially select data and provide decision support.
- *Information Module* - includes the requisite data and business rules to support the mapping, decision parameter calculation, and decision support functions in the form of a Data Container.

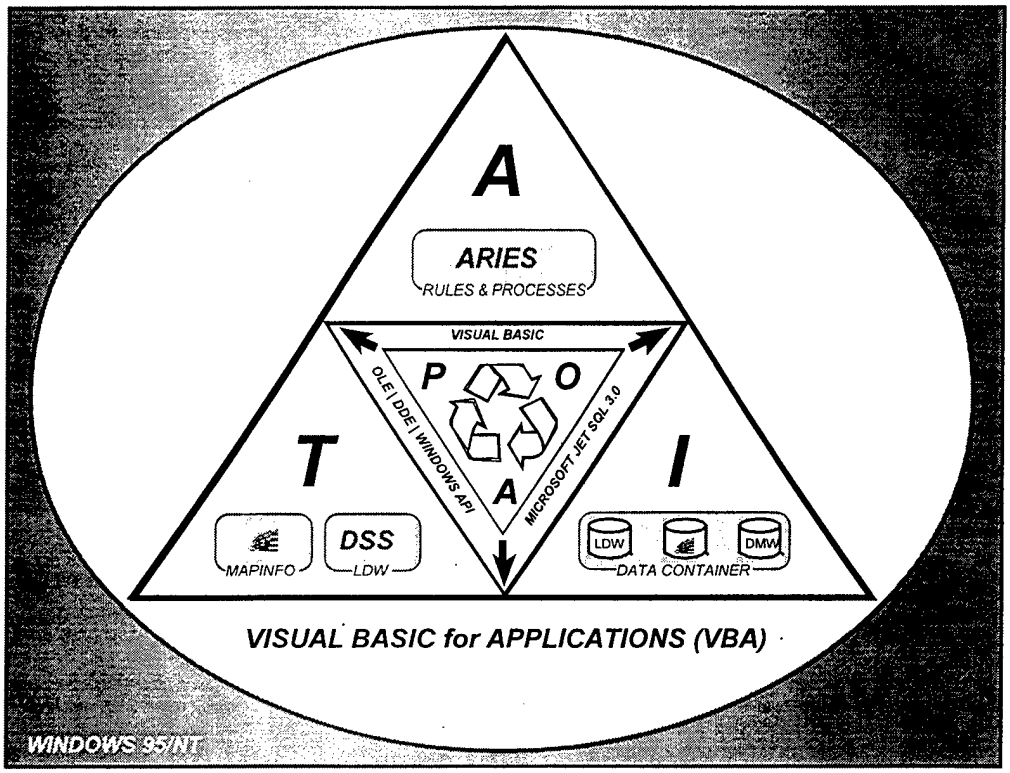


Figure 3-6. ARIES Baseline Infrastructure Archetype (BIA) Model.

Because the movement of data between USARC legacy data sources and SDSS modules has the largest data flows, the development team implemented a container approach to manage all data flows within the context of the macro-architecture (Figure 3-7). The Data Container represents an abstracted data repository that contains all the meta-data, source data and COTS support data necessary to the relocation decision, and provides a means for ARIES to be portable for mobile users and to operate independently

from the USARC LAN for networked users. Source data is migrated to a separate database structure (data migration warehouse — DataDepot) for the following reasons:

- Provide an accurate, centralized, and high quality — data repository for calculating decision parameters.
- Principle of minimality - only use whatever data is required to support reserve unit site selection.
- Accelerate data exchange between modules by using the native DBMS format of the USARC infrastructure.

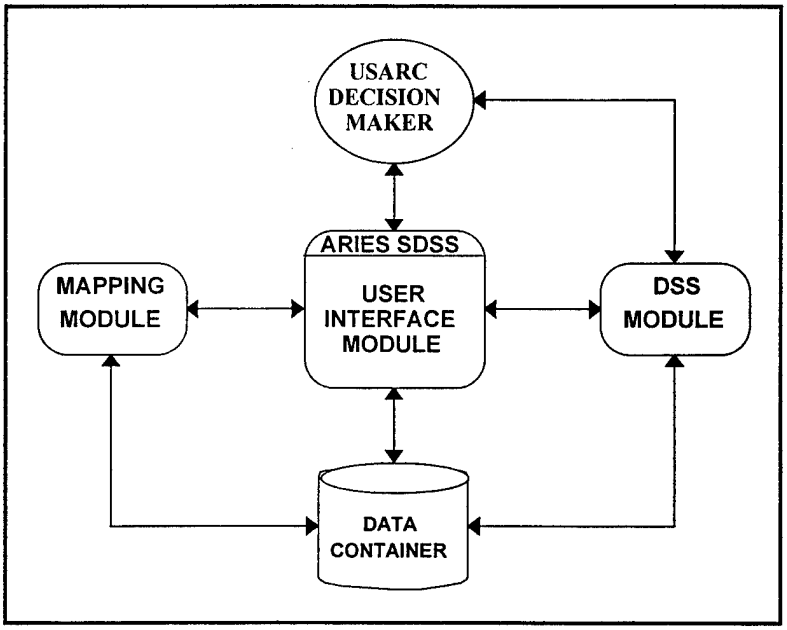


Figure 3-7. ARIES Macro-Architecture Diagram

b. ARIES Micro-Architecture Stage

After considering a number of candidates and conducting an infrastructure review; the development team identified the primary technologies that fit the organizational objectives, with which to populate the macro-architecture (Table III). The information within the Data Container component represents the migrated USARC source data necessary to support the relocation decision analysis (DataDepot); the MapInfo data files necessary to support the display of maps and geocoded information;

and the ARU-DM master templates to support the importation of computed decision parameters.

Macro-Architecture Module	*Micro-Architecture Component*	*Reasons for Selection*	*Communication Protocol*
ARIES User Interface	Visual Basic 4.0 Application	USARC Requirement	Various
ARIES Business Rules	Visual Basic 4.0 Application	USARC Requirement	Visual Basic for Applications (VBA)
Mapping Tool	MapInfo 4.0	USARC Requirement	Microsoft OLE
Decision Support System	Logical Decisions for Windows.	Only DSS COTS product capable of auto-weighting.	Windows API Routines and Macro scripts.
Data Container (DataDepot)	Microsoft Access Database	Performance gain from using Native DBMS vice Third-Party (ODBC)	Microsoft JET SQL 3.0

Table III. Macro and Micro Architecture Comparison Chart.

Interoperability between components can be achieved by a variety of means, but it must be delineated in the architectural framework and applied consistently throughout the application. In the case of ARIES, many various communication protocols had to be used to ensure proper component iteration. For example, because the LDW application was originally designed for a 16-bit Windows environment, it was incapable of being controlled via established distributed computing standards (e.g., OLE). Therefore, the development team used intrinsic Windows Application Programming Interface (API) routines to manipulate the LDW window displays and the Visual Basic SendKeys Function to process macro-scripts. In short, the ARIES U/I controls LDW by simulating a user's input.

c. *ARIES Domain Prototype*

Although each of the technologies brought strengths that USARC wanted to take advantage of, integration of the components into a coherent process that could be controlled from a single point soon became a formidable challenge. The developers were faced with the difficulty of minimizing the coupling and data transfer between the major components in order to maximize application performance. The first step — was for the development team to fully understand all of the components — including their design intentions, capabilities, strengths, weaknesses and external communication protocols. The second step — was to partition the U/I views of the analysis prototype into smaller segments and put the riskiest parts at the top of the prototype list, rather than letting them slip to the end. This "Piece-of-the-Pie" or subset approach, allowed the developers to bind each U/I view in turn to the applicable component (Table IV).

By using RAD prototyping, the development team was able to build the interface quickly and easily using a visual development methodology (VDM), and then using component-based development, to link the U/I to existing data sources, infrastructure and components to ensure connectivity. Later in the Implementation Phase the team would place the appropriate code for business rules and data manipulation behind the approved user interface. In this manner, the abstract interface encapsulated, or hid, the actual system implementation from the end-users. The decision-maker can only view and access the system by way of this single point interface.

User Interface View	*Micro-Architecture Component*	*Communication Protocols*	*User Inputs*
• Map Select Tab • Map Query Tab	• MapInfo 4.0 • MapBasic 4.0	• OLE; Dbase III • OLE; Dbase III	• Graphically manipulate MapInfo tools (zoom-in, zoom-out, select, grabber)

Information Panel	MapInfo databases USARC databases (GeoCoded) DataDepot	ODBC (FoxPro) Dbase III	Scroll UP/DOWN Accept Facility Info
Facility Selection Panel	DataDepot	JET SQL	Moving Unit UIC FacID (Qty 1-4)
Control Panel	ARIES U/I	Visual Basic for Applications (VBA)	Start ARIES Reset ARIES Exit ARIES

Table IV. ARIES User Interface/Micro-Architecture Component Binding.

d. ARIES Visualization Artifacts

Before proceeding to the implementation phase, another formal walkthrough review was conducted with all involved to ensure that the proposed procedural flows, component architectures, and U/I prototype were practical, close to what the user expected, and adhered to specified standards. Figure 3-8 illustrates the final approved component framework solution.

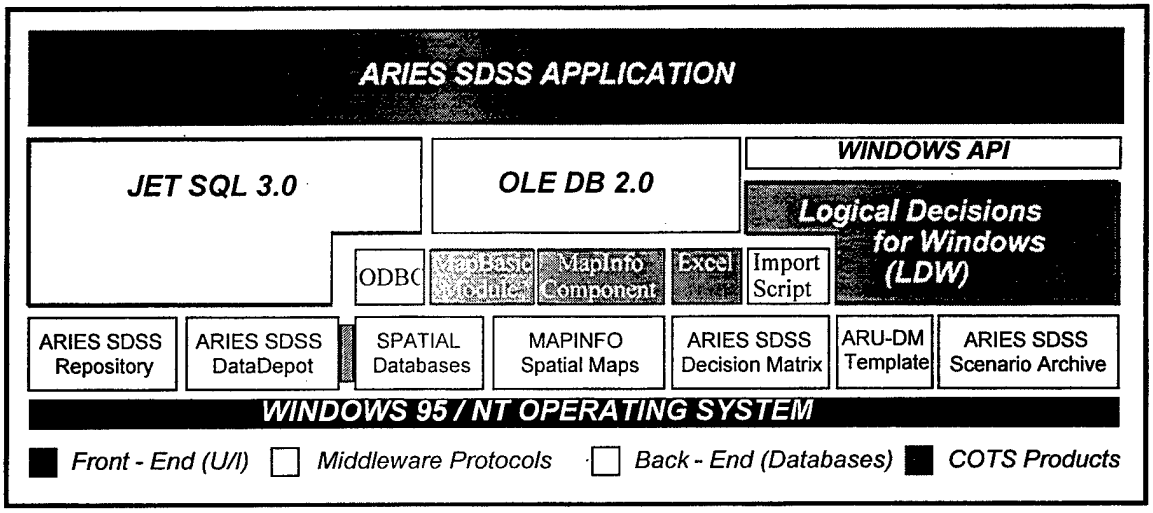


Figure 3-8. ARIES SDSS Component Framework Solution.

4. Implementation Phase

At this phase in the C2C process, the component architectures and user interface prototyped in the Visualization Phase are translated into source code in 3GL or 4GL syntax. If an RDBMS is involved, the database schema will generated and integrated with the final application. The purpose of the coding is to translate the detailed architecture design into computer language suitable for the designated operating environment.

Prior to this stage the development team "must first agree on the processes and methodologies it intends to follow, and it must decide which of those processes are going to be followed religiously -- and which ones will be honored in spirit, but perhaps not to the letter of the law" (Yourdon, 1997). Once this has been decided, the tools and technology can be chosen — or rejected — accordingly.

a. ARIES Development Methods, Techniques, and Strategies Stage

Once the application has been designed in the form of interacting components, it becomes very easy to add new ones and scale up the functionality of the system or adapt the system to meet a design constraint. This was the case in ARIES for transferring the computed decision parameters to the decision support system. Because the LDW DSS is a 16-bit Windows application, it was incapable of reading Microsoft Access database files without a custom modification by LDW's development team. The solution was to convert the Access data table to a format that LDW could import reliably. Component testing of the LDW DSS indicated that a Tab-Delimited format was the most suitable and that by using the Microsoft Excel spreadsheet component available in USARC's infrastructure software suite, the information could be converted and imported quite easily using OLE communication protocols.

Another advantage of the approved component framework, was that once all the required components and interfaces had been specified, many of the coding activities could be performed in parallel as long as the input data necessary for the component was available.

> Once the allocations of system requirements to...software components has been made, parallel...software definition, design, and implementation activities are initiated. During these parallel efforts, the...development team must continuously cooperate and communicate to ensure that their implementations are compatible. This continued interaction is critical to the successful integration of the...software components. (USAF-STSC, 1992, p. 20).

b. ARIES Coding Stage

The ARIES U/I and application was coded using the Visual Basic for Applications (VBA) language and standard structured programming techniques. Only one form was created during the Domain Prototyping phase which contains all the VBA objects and the integrated MapInfo object of the ARIES U/I. The rest of the partially-compiled application is comprised of several VBA-coded files and one MapBasic program (Appendix D):

- SUBMAIN.BAS - Initializes the ARIES U/I; Loads and connects COTS components to U/I; Displays the U/I.
- UserInterface.FRM - Contains all procedures, business rules, and logic for controlling the ARIES U/I.
- PublicDeclarations.BAS – A library module that initializes all public variables and constants; Contains the variables that represent the SQL business rules for calculating the ARU-DM decision parameters.
- Library.BAS - A library module that contains all the procedures called by other procedures more than one time.
- OLE Library.BAS - A library module that contains all the procedures supporting OLE communications between COTS components.
- OLE Callback.CLS - An object class description necessary for communicating with the integrated MapInfo component.

- MapBasic.BAS - A library module that initializes all public variables and constants for communicating with the integrated mapping component.
- ARIESArch - A MapBasic program that spatially selects the desired data within 50 miles of the specified location and locates the closest Army Reserve support facilities.

Code evaluation occurs during and upon completion of coding. It ensures that coding techniques are acceptable and that standards have been followed during the coding process. The evaluation was conducted by the development team and NPS advisors, using peer-review and informal walk-throughs in order to verify the code against the design, and to ensure compliance.

c. ARIES SDSS Application

The gain in performance of this spatially enabled DSS over its manual counterpart is formidable. A process that previously required weeks, can now be completed in minutes. The SDSS DSSA is very flexible and offers numerous ways of eliciting preferences and displaying results. The structure is adaptable enough that it can be easily expanded to include additional decision parameters, criteria, and improved decision models.

Given the large number of components that must be centrally controlled by the U/I, and the intense database activity involved in calculating decision parameters, a local installation of the ARIES application at USARC Headquarters was recommended. The development team believed it was critical to perform this type of testing at the actual client site in order to best measure the application's performance in an operating environment that could not be simulated at NPS. Since this was economically infeasible, the application was rigorously tested on the provided laptop computer, and the burden of network "beta testing" of ARIES was shifted to the USARC staff.

D. SUMMARY

Our experience in using C2C to help USARC develop the prototype and test the SDSS concept, indicates that this approach can have a major impact on the way users and developers collaborate on producing functional software. C2C's structural approach and use of component based development techniques to incrementally build and deliver mission-critical applications, formalizes a process that was, until recently, overly complex and unwieldy.

The C2C process puts an early emphasis on addressing a concept's high-risk areas by rapidly prototyping a proof-of-concept version of the intended system that defines its specific component architecture. It does not assume a fixed set of user requirements at the concept inception, but allows for iterative refinement of the requirements as the project evolves. Changes are expected and accommodated. The main focus still remains the final software product and its quality, as measured by the degree to which it satisfies the user's expectations.

As component-based development becomes more widespread and focuses on architecture, the complexity of the distributed operating environment should wane. Until that occurs, component-based systems built by non-programmers will probably continue to be interconnected in an unstable "hodge-podge" fashion. The best way to improve the integration of these systems is to adopt an enterprise-wide, integrated architectural strategy that incorporates standards that act to limit the inherent complexity faced by users and developers. In this sense C2C acts as a technological "gap filler" between baseline infrastructure and target applications.

As with any new software development process and modeling methodology, nothing works exactly right the first time out. The USARC-NPS innovative collaboration was no different. What appears to occur smoothly throughout the development of the

ARIES SDSS application, in retrospect was actually disjointed and haphazard as USARC and NPS personnel underwent a "learning experience" about merging component-based technologies with architectural thoughts and ideas. These "lessons-learned" are what the next chapter is devoted to exploring in detail.

IV. LESSONS LEARNED.

"There is a point at which methods devour themselves."
Frantz Fanon, *Black Skins, White Masks*, 1952.

"A tool knows exactly how it is meant to be handled, while the user of the tool can only have an approximate idea."
Milan Kundera, 1978.

"Reality is the murder of a beautiful theory by a gang of ugly facts."
Robert L. Glass.

"The only thing new in the world is the history you have not learned."
President Harry S. Truman

A. OVERVIEW.

This Chapter examines some of the most critical lessons that the development team learned while constructing the ARIES application. It is by no means inclusive of all the lessons learned during the project, but rather a representative sample of the lessons related to component-based development. Each chapter section is presented in categorical form, beginning with users and analysis; application development; rapid prototyping, and component-based development. Each of the pertinent lessons related to a particular section is stated first and then discussed at length.

B. CONCEPTUAL ANALYSIS AND THE USERS.

- ***Because of software development's inherent malleability, it is difficult to anticipate which requirements will be implemented easily and which will decimate the project.***

Deciding what to build and what not to build is the age old dilemma that faces all developers. Experience has proven that software requirements are the biggest impediment to properly engineering complex systems. Conventional as well as state-of-the-art software development practices have failed to make appreciable headway against

persistent and pervasive problems with requirements management (DON-NRL, 1996, p. 1). Beyond this need for discipline, requirements are intrinsically hard to define well. In an attempt to better manage requirements analysis, the development team countered with the Conceptualization Phase of the C2C Heuristic. By devoting crucial time and energy at the outset, we hoped to capture, structure, and contain the sheer essence of the project. This could not occur without an equal give-and-take on the part of the intended users. Through the formation of a collaborative effort, whereby the group built a little and prototyped a lot, a productive synergy yielded stable requirements from which to design architectures and model the problem domain. The users specify the "what" of the problem and the developers focus on the "how" of the solution. This iteratively and multiplicatively leverages the users' business knowledge in concert with the technical acumen of the developers. Using prototyping as the practical solution The cognitive layers and the feedback cycle of the C2C structure acts to progressively guide the collaborative ensemble en masse from high levels of abstraction to a specialized solution space.

In the case of ARIES, the characterization stage and U/I prototypes were the most instrumental in building the application correctly the first time instead of enduring endless cycles of change management. The developers were able to understand exactly what the SDSS needed to perform, were able to communicate side issues and discrepancies in the language of the users, and used an architectural focus to provide a means for controlling the production of the final application (DON-NRL, 1996, p. 6).

- ***The developer cannot understand the business rules and processes well enough.***

Most application generation languages do a good job effectively designing the user interface and providing the underlying support code to connect the U/I to the pertinent database(s). But it is the business rules and data processing logic that are the

heart and soul of a successful application and are the most difficult to automate. It was this understanding of the rules to which the NPS development team devoted a majority of its analysis efforts. By focusing early in the Conceptualization Phase on thoroughly comprehending and characterizing the exact rules and processes used by the USARC decision-makers, the team avoided extensive and unnecessary rework later in the implementation stages. Using process roadmaps and documented business rules, ARIES development became a straightforward exercise in prototyping the U/I, identifying the required SQL database operations, linking the Rules to the U/I, and linking the rules to the SQL.

- ***Keep the user engaged in the collaborative process.***

ARIES's intended users failed to adequately specify top level system requirements as quantitatively as possible, and developers failed to better elicit them. Too many trade-offs and design decisions were left to the developer. Conflict at structured walk-throughs occurred between the developer's view of the design and the users' unarticulated view. Large amounts of development time were lost in several C2C stages because of these discrepancies.

Collaboratively prototyping in one room or meeting is difficult enough under the best conditions, but in the case of ARIES the development team resided at NPS in Monterey, California, while a majority of the time the users were at USARC headquarters in Atlanta, Georgia. Attempts to regulate the collaboration using email proved ineffective. A second problem in this arena involved the USARC LAN network software upgrade from Windows 3.1 to Windows NT which was not scheduled for completion until after product delivery. As a result, incremental prototype builds could only be alpha-tested by USARC decision-makers during the monthly progress meetings. Instead of stabilizing the design, the opposite outcome occurred as the prototype demonstrations

elicited user suggestions not anticipated by the developers. For example, the user wanted to be able to select the facilities for site relocation graphically using the mapping component. The development team mistakenly assumed that the map view should label the selectable nodes with its corresponding facility identifier (FacID). The Domain Prototype was built around this misconception, only to discover at the prototype demonstration that users had trouble "drilling down" through the map layers of the map view because they were in the habit of selecting facilities by the closest cities and towns rather than by FacID.

- ***Identify who is responsible for software maintenance prior to development.***

One the biggest mistakes that can occur with regard to software development is to delay the selection of the software maintenance organization until after the development of the software application and data structure has begun. The developer may have deployed a superior application, but if is incompatible with existing infrastructures or unsupportable by the recipient organization's IS support staff, then the program probably will not be used and likely "wither on the vine" from lack of support. The solution is to select the provider of the technical support prior to the commencement of application development. The IS support can be provided in one of two forms:

- Externally by the software development team/organization, or,
- In-house or organically (i.e., by an organization of the Army).

If an organic organization is selected, then the software tools and applications used for development must be provided to, and agreed upon, by the maintenance organization. In the case of ARIES, the initial Analysis Prototypes were developed using Borland's Delphi 2.0 fourth generation language. Although the lead developer had experience with Delphi client-server environments, the USARC IS Staff were inexperienced and unfamiliar with the Delphi programming environment. Therefore

Delphi was summarily vetoed by the USARC IS Staff and two additional requirements were imposed in the second round of the Conceptualization Phase:

- Both the ARIES Application and Administer must be developed in Microsoft Visual Basic 4.0.
- All Visual Basic and MapBasic source code must be provided separately to USARC in an uncompiled form.

This resulted in a six week loss of development time while the development team attended several professional Visual Basic workshops and redesigned the Analysis Prototype to comply with these new requirements.

C. ARIES APPLICATION DEVELOPMENT LESSONS.

- ***Small development teams are best.***

It is a fundamentally simple principle that if you want to limit the number of defects in a project, then limit the number of developers involved in producing the project. Good people produce good software (Yourdon, 1996, p. 28). The results from using a small team for the ARIES SDSS were better management, less coordination, and improved communication between both team members and users. By using a design heuristic, automated tools, and architectural structure, the development team was able to remain small in size (3-8 members), and reducing the amount of code to be written and tested. Small teams using modern tools, techniques, and user experience can achieve high levels of productivity.

- ***Spatial selection, when properly implemented, can substantially reduce decision model parameter calculation response times.***

During the Visualization Phase, the domain prototype of the ARIES SDSS performed much more slowly than the collaborative team had anticipated. Some of the problems occurred because of overhead from complex SQL queries between the U/I and

the RDBMS. For example, calculating the available MOS's from the closing Army Reserve units requires the following logic:

> ***Determine all the individuals assigned to Closing Units who live within 50 miles of the Proposed Site having MOS's matching a prospective Moving Unit's requirement for a particular MOS.***

The example took nearly an hour to execute on a Pentium 120 MHz computer, regardless of the population density of the location. Even with performance tuning execution time was still about 40 minutes on average to perform the query. The solution was to utilize the specialized geographic processing capability of the MapInfo component to spatially select only the database records within the 50 mile geographic boundary of the proposed site. This required obtaining a waiver from USARC to allow the development team to geocode the Army Reserve Unit personnel database (G18CWE) using the MARS Engine, to migrate the database, and to spatially enhance it and two others for geoquerying by ARIES. The performance gain from this technique was on the order of five to seven times faster (Figure 5-1).

- ***Avoid design error propagation by adhering to a philosophy of "Daily Build and Smoke Test."***

When discussing software development in terms of incremental prototypes or iterative "builds," one thinks of monthly or weekly milestones. This correlates to the business world and our daily lives as a "normal" production pattern. The trouble lies in the rapid changes that prototyping perpetuates; factor in concurrent development by team members, and you have a recipe for disaster when project assembly occurs in one month (e.g., "let's put it together and see what works").

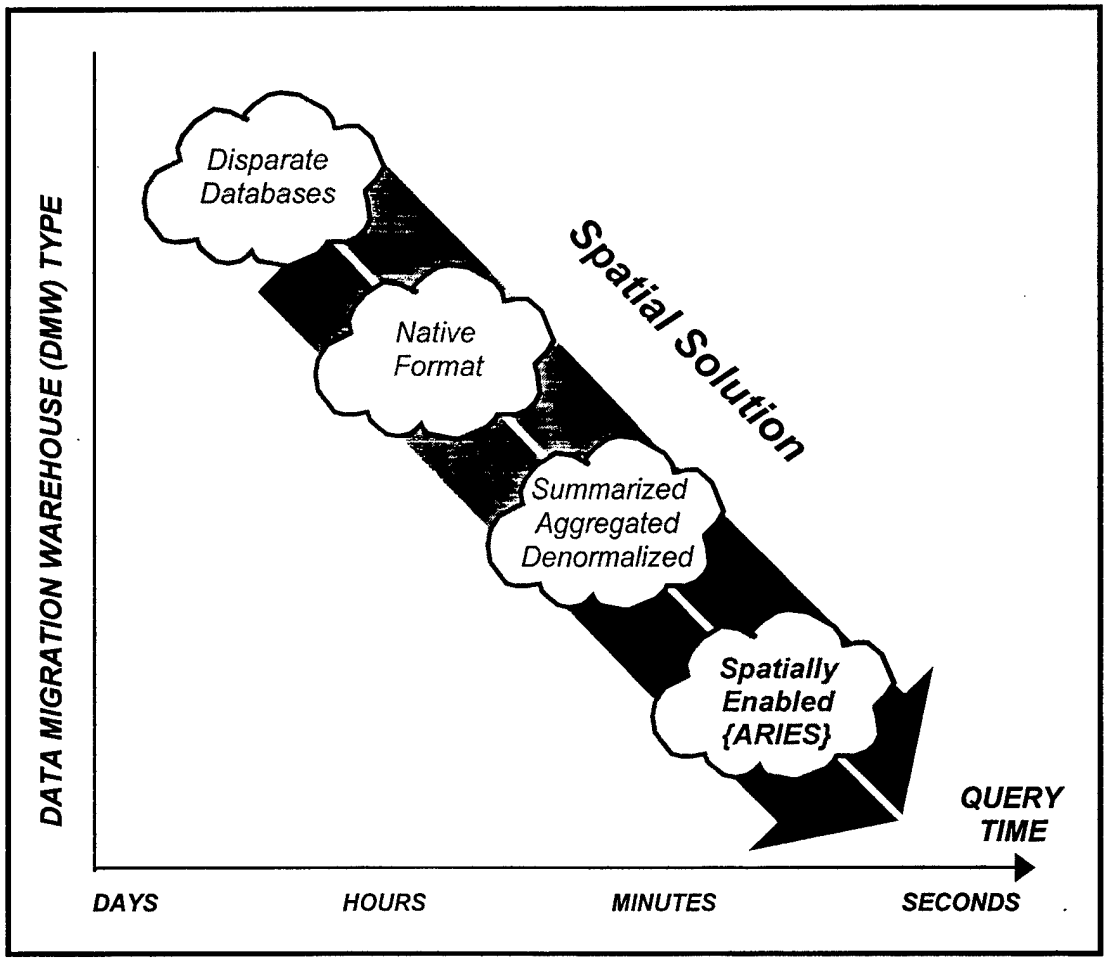


Figure 4-1. Spatial Solution Diagram.

Component-based development allows the developer to follow a different approach — the *Daily Build* and *Smoke Test.* This process can be used effectively for projects of any size or complexity and consists of developers, on a daily basis assembling, compiling, linking, building, etc., all the project components and code modules into a deliverable application and then testing to verify its basic operations. The team is not allowed to continue coding until the current "build" passes satisfactorily, and component failures can be attributed as the responsibility of the component creator. In this manner the developers start every day with a product that functions correctly, and

ensures that any defects that are present were induced within the last 24 hours. It is far easier to retrace an individual's one day effort, than a team's monthly effort. This prevents developers from unwittingly developing incompatible components and corrupting the overall design.

Why is this so important? The process pays for itself by reducing the time-consuming development risks of low quality, integration failure, and poor progress visibility (McConnell, 1996, p. 405). Not only does the developer always have a deployable application, but the users, project managers, and other team members always have a daily status of what is complete and what needs to be added. Any problems which arise tend to be relatively small in nature and can be dealt with accordingly, rather than shifting into "crisis mode" to rework code following a monthly or quarterly project build. "The daily build is the heartbeat of the project. It's how you know you're alive" (McCarthy, 1995).

- ***A strong, stable, and well-defined architecture prevents degeneration of design.***

It can be extremely difficult to maintain an application architecture as the development process unfolds. Over time, performance and delivery pressures can combine to blur the architectural design, resulting in unwanted degeneration of the structure at all levels. A "good enough" architecture may be the result of attempting to accommodate a given COTS component structure, but a standards-based architecture is critical to the development process. Quality architecture is more likely to influence a successful project than a given set of components. Component technology and good architecture need to be intrinsically related, otherwise architectural mismatch can occur (Garlan et. al., 1995, p. 18). Rapid prototyping can accelerate this mismatch unless a strict development discipline is enforced by the developers to rework and revisit all affected components after each prototyping cycle ends and during component integration.

A well-thought out architectural solution can be effectively and efficiently implemented in almost any language, environment, and infrastructure.

D. RAPID PROTOTYPING LESSONS.

- ***Decide up front whether the prototype is to be a "throwaway" or a full production system.***

A throwaway prototype or scaled-down model is used for feasibility exploration and proof-of-concept studies in which user requirements are incomplete and the system approach is uncertain. However, if a prototype is being constructed using known requirements, and only minor uncertainties remain regarding the system approach, then an evolutionary prototype can be built, in which some or all of the prototype is retained (Gordon and Bieman, 1995, pp. 85-86). Usually the throwaway is built first, with the developer proceeding quite rapidly as the user requirements are defined and the system architecture is confirmed. The downside is that, in its limited, scaled-down form, the prototype is usually incapable of being deployed as a full production system because so many shortcuts have been taken to illustrate whether something could be done. When the design specifications are complete, then it can be rewritten to fulfill a more flexible, scaleable, and extensible form.

Originally ARIES was budgeted for a two-phased development: a stand-alone throwaway prototype, and a full production network-centric version. The network model was to be a three-tiered, fully object-oriented client-server application that would be Windows NT enhanced to maximize decision parameter calculation using remote OLE automation. Difficulties with this scheduled arrangement occurred when project funding was reprogrammed halfway through the C2C development cycle. This required a collaborative reorganization of the ARIES SDSS to enable the application to execute in the network environment without being optimized for it.

Conventional wisdom counsels that whether "[you] plan to throw one away; you will anyway," (Brooks, 1987) but often, in the interests of time and expense, a prototype will be retained in some form from the start. As developers we are not advocating one particular method over another, only that in the best interests of the collaborative design team, it is imperative to decide from the outset which one it will be.

- ***The collaborative nature of prototyping requires more of the user's time than he plans for.***

The challenge of managing a prototyping effort can be much more complicated than managing conventional software development. The difference stems from the heavy emphasis placed on providing the users what they want and need. This leads to modeling processes in new ways, which in turn requires near constant feedback to prevent wasted effort and trivial pursuits of unintended features. Sometimes this constitutes more feedback than the user is capable of providing.

Because USARC could only review U/I prototype once a month during their trips to NPS, user feedback tended to come very late in the prototyping cycle. This latency meant constructive feedback came at a time when incorporating the changes incurred significantly higher costs. Changes in the user interface prototype at the end of each Visualization Phase walk-through twice resulted in an additional four weeks of design, testing, and unit integration.

- ***Quick visual results encourages the user(s) to suddenly see possibilities where before they had only seen barriers.***

A striking phenomenon about visually-oriented prototypes is the uncanny ability to elicit Archimede's "Eureka!" reactions from the pool of intended users. The potential end-user goes from a position of "I'll know it when I see it," to "if you can do that, then surely you should be able to do this?" It provides a sense of empowerment to the user and allows him to shape and envision the final system., In practice however, this can only

occur after he has gained some experience from the system proposed by the developers (Gordon and Bieman, 1995, p. 86). As with every new unprecedented capability, there comes a dark side. The user's hunger for more and more functionality can become insatiable without some outside restraint imposed by the development team. This may result in uncontrolled "requirements creep." Because the prototype can so effectively and easily mirror the final system, the intended user(s) may have little sense of the relative costs and the diminishing marginal returns of adding functionality beyond the scope of the initial concept. In short, "perfection" becomes the enemy of "good-enough."

In this sense it is the duty of the developers to guide and assist the users in prototyping the final system around a reasonable bandwidth of the users' stated goals. Their experience and objectivity are critical in knowing when to stop prototyping and start coding. Otherwise excessive gold plating may result in a great system that never gets built.

- ***Using COTS products does not eliminate software maintenance.***

On the contrary, because the origin of installed components is external to the developers, software maintenance plays an even larger role than it does in conventional software product development. If a component is modified and the vendor releases a bigger and better version, as they always do, who then is responsible for the care and feeding of the application? The vendor? The development team? The recipient organization? Secondly, if the developers use the product "as-is" and the vendor updates other components in the application architecture, who bears the brunt of the burden to ensure component compatibility? Each change in the implemented design requires a new version of the system to maintain the technical integrity of the application. Otherwise the incompatibilities become cumulative and propagate instability throughout the component architecture.

This was the case with LDW, the ARIES DSS component. As the other components received major product upgrades, including the operating system, the LDW DSS remained unchanged. Because of the unwillingness of the LDW vendor to migrate LDW to a 32-bit application, the burden of maintaining the ARIES product over the next 18 months will become even more difficult without radical work-arounds to make a 16-bit program backwards compatible within a 32-bit component framework. As a result, the viability of ARIES will become less and less certain as USARC upgrades its infrastructure.

In summary, you cannot simply build a component-based application and then forget about it. The more easily such an application is built, the more critical the maintenance effort needed over the application's lifecycle.

E. COMPONENT-BASED COMPUTING LESSONS.

- ***Building custom components is really, really, hard.***

Do not be fooled by the component evangelists. It is very difficult and time consuming to build even simple components that work and communicate well with other components and applications let alone complex ones. In today's dynamically complex event-based development environments, a developer must be able to anticipate every possible use and interaction a component could be expected to fulfill and undergo respectively. Many component building and application software packages have been created to address these issues, but testing components in infinitely diverse combinations is a Herculean undertaking. Further, using contemporary methodologies does not insure success; many dismally poor components have been produced using modern tools and techniques. High quality and reliability comes at a premium cost. It is far easier to buy and modify a well-designed COTS product than to develop and validate a custom built one. The challenge in doing this well is to know and understand thoroughly the available

off the shelf commercial products, because many of the problems encountered during implementation simply cannot be determined before integration begins.

- ***The learning curve for mastering component-based application development is steep.***

It is important that, prior to using a particular component or fourth generation development language for developing a component-oriented application, the developers are comfortable and familiar with the product. One person cannot be expected to instantly master several different components well enough to be an effective and efficient developer while at the same time attempting to prototype the system. "Just in time learning" is insufficient to guarantee a stable design and is often a blueprint for disaster.

- ***Component communication protocols are the weakest link in the development chain.***

Some components are capable of using open or de facto standards (e.g., OLE), some must utilize proprietary communication protocols (e.g., DDE), and others cannot operate with various components at all. In order to effectively implement the ARIES component framework, the developers were forced to use a combination of all three methods to achieve the desired functionality. This approach was not without its drawbacks, for the less 'open' the protocol, the less functional the implementation. For example, because LDW is incapable of importing most current file formats, Microsoft Excel had to be used to save the resultant scenario's decision parameter matrix in a tab-delimited format that LDW could process; Visual Basic could not do that easily. This introduced another component in the architecture whose only function was to serve as a file transfer agent, a function which is not the component's strongest suit.

The MapInfo component, on the other hand, fully embraces the de facto OLE standard and effectively documents the component's interconnections. As a result, approximately ninety percent of the functionality within the MapInfo application could be

done within an integrated map window. To similarly manipulate the LDW application, the ARIES U/I hypnotized it into executing user-simulated keyboard commands using Microsoft Windows API routines and macro scripts. Essentially the ARIES U/I tricked the program into thinking a user was present when he was not. This technique is strictly limited in that only about 20% of LDW's capability could be controlled in this way by ARIES.

- ***Use of Commercial-Off-The-Shelf (COTS) components can result in "dead code" solutions.***

In some cases the use of an appropriate or inappropriate component can result in some or most of the component's functionality not being used (dead code). Essentially a large component with many capabilities may only be needed to provide a single function or a small subset of the original functionality. An example of this is the use of Microsoft Excel, a large and complex application for spreadsheet manipulations, which is used exclusively in ARIES to load the decision parameter matrix and save it in a different format. Similarly, MapInfo is used only for geoquerying and facility selection, in spite of its extensive thematic mapping capability. The resources that the mapping component consumes (e.g., 4 MB of random-access memory) are lost to other applications when only 15% of the component is used.

- ***COTS components are not all "open" standard compliant.***

This was probably the hardest lesson the ARIES development team had to contend with, and is a major issue for anyone interested in pursuing component-based development as a solution provider. In the past components or small, specialized 16-bit applications were designed and built using proprietary standards combined with programmers' idiosyncratic methods to achieve a certain level of performance. As operating systems have matured and become more sophisticated, components and

programs are able to take advantage of object libraries and infrastructure to achieve functionality. Yet openness is still measured by proprietary designs and consumer acceptance. Providing a uniformly understandable means for accomplishing tasks is key to limiting communication chaos and reducing risk within the application's architecture. Without standardization and architecture, the process of integrating unique application components becomes extraordinarily complex. Since no single vendor has accommodated the entire range of infrastructure management needs, de facto standards have emerged.

This is where the real trouble arises, for third-party components and updates from vendors are not exactly 'plug and play', and frequently have a tendency to be falsely documented. The integration of various component data formats and communication protocols can be very complex, and subtle "gotchas" within a component may render its role in the component framework null and void. Initially the development team tried to grab too much "off the shelf" and hastily put it all together during the prototyping stages. This yielded mixed results — mostly bad. The team was so enamored by the prospect of a buy-and-integrate strategy that it failed to realize that some COTS products can only interact in accordance with its own set of standards, architectures, and design. Not all COTS products are universally pliable. They may require in depth analysis of their intended operating environments and structure.

As an example, LDW is capable of screen capturing its various graphic analysis displays to the clipboard or a graphic file. The undocumented difficulty is that the "clipboard" was implemented by the programmer in C++ without using Windows native API routines. Additionally, the graphic file is the 16-bit Windows meta-file standard (WMF) that is supported by only a handful of programs (e.g., Microsoft PowerPoint). Therefore, the only method available to allow the user to personalize the standard reports

feature of the ARIES U/I was to implement another large "dead-code" component, like PowerPoint, to once again act as a file transfer agent for graphic analysis editing.

Another difficulty emerges when trying to integrate new versions of existing components within the application's architecture. The new version may have a different look-and-feel and require more memory, and/or previous data formats and protocols may no longer be supported. In any case, another C2C incremental cycle will need to be completed to ensure application compliance and functionality.

The only way in which to make various disparate components operate together reliably is to adopt a strong single vendor infrastructure solution (Microsoft) and work within the confines and limitations inherent to the selected standards and architecture. Hopefully in the future, emerging software technologies will be developed and distributed as a new kind of component. A component that is functionally cohesive, based upon open standards, small in terms of complexity, and therefore capable of operating in conjunction with all other possible components and software applications. (Figure 5-2).

F. SUMMARY.

It is becoming increasingly clear that software development, which relies heavily upon component-based architecture, requires modifications to the traditional development methodologies. Although some of the lessons derived from ARIES were recurring themes of software developments lessons of the past recast in component-based form, others were unique to the particular technologies and methods used within the structured C2C paradigm. In the final chapter, we summarize the lessons learned from this SDSS exercise, and project how these can be adapted to future component-based development applications.

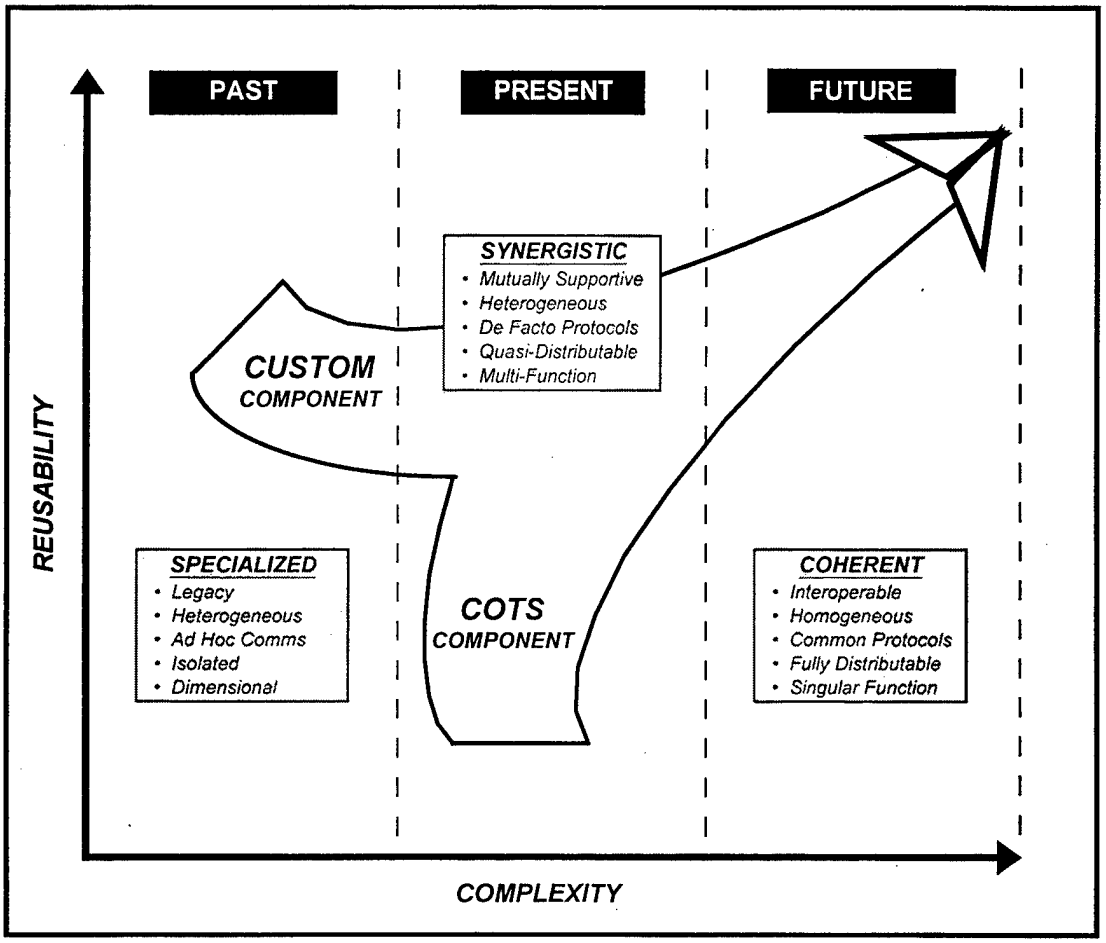


Figure 4-2. Component Evolution Diagram.

V. CONCLUSIONS AND FUTURE RECOMMENDATIONS.

"*Institutionalizing organization-wide decision-making processes and an architecture is the key to all of our information system development efforts...and our primary measure of success is impact on the bottom line.*"
Chief Information Officer (GAO, 1994).

"*It is generally very difficult to keep up with a field that is economically profitable, and it is only natural to expect that many of the techniques described here will eventually be superseded by better ones.*"
Donald E. Knuth, *The Art of Computer programming*, 1973.

"*We build systems like the Wright brothers built airplanes — build the whole thing, push it off a cliff, let it crash, and start over again.*"
Professor R. M. Graham, *Software Engineering*, 1969.

A. INTRODUCTION.

This chapter synopses the underlying themes characterizing the development and implementation of the ARIES SDSS. It summarizes our experience with using the Concept-to-Code (C2C) design heuristic to construct software within the context of a new structured paradigm, and provides suggestions and insight into the ARIES Next Generation (NEXGEN) prototypes.

B. LOOKING BACK.

Reflection upon the ARIES SDSS implementation could be perceived in one of two ways: as "blind luck", a one-time-success story that by sheer happenstance delivered exactly what the user wanted with a development team that had no prior experience with the underlying development tools; or, in a more realistic light, as an application which provided a means to explore, justify, and demonstrate the validity of using component-based computing in conjunction with rapid application development techniques, to

quickly and accurately produce a scaleable, extensible, and useful decision support application.

The C2C design heuristic employed for ARIES allowed the collaborative team to quickly conceive and sketch screens and displays, and then build the user interface up front in a short turnaround period. Defining how the components of the application architecture were to communicate with one another, drove how the intended functionality of the architecture was to be accomplished. The heuristic allowed the developers to "architect" the system into small components and put the high risk segments (e.g., primary functionality) at the head of the prototyping line, rather letting them slip to the end. It was important to build the most difficult components first and then add the "nice-to-haves" later on. C2C also greatly improved management oversight by allowing the application development process to be layered from a generic concept to specialized solution.

In summary, the C2C process has resulted in a number of benefits:

- Facilitated understanding of the development process by managers and programmers using a one-page graphical representation.
- Provided concise, definable phases with clear entry and exit criteria and staged iterative development.
- Fostered an atmosphere of continuous quality process improvement between users and developers.
- Allowed small development teams to implement large scale, complex applications in short development cycles.
- Provided clear communication of project status between the development team and management on a daily basis.

The success of the ARIES SDSS could be considered surprising, but in reality it was no fluke. The C2C process, applied carefully, can lead to powerful applications within a surprisingly short period of time, months vis-à-vis years as in the case of ARIES.

Undoubtedly, C2C will continue to undergo experimental refinement and improvement, as it may very well become the mainstay of software application development.

To date, the ARIES SDSS is effectively deployed on the USARC local area network, as well as USARC staff officers' laptop computers. The results of the ARIES SDSS deployment have been very positive for both the NPS development team members and the USARC staff. The issues of timely and consistent response to site relocation questions appear to be more than adequately met by the ARIES SDSS prototype.

C. LOOKING FORWARD.

1. Current Technology Options.

a. Infrastructure: A Component Update

In the short time since the deployment of the ARIES SDSS prototype on the USARC LAN, several changes have occurred with its primary components. All of the COTS infrastructure components, with the exception of Logical Decisions for Windows, have had new versions released by their associated vendors. Each component's functionality has been greatly enhanced and conforms to widely accepted client-server protocols. This includes the Visual Basic development language, which has been significantly upgraded to function exclusively within the component-based paradigm. Applications created in Visual Basic 5.0 are now fully compilable and the ODBC component has been integrated into the new JET SQL 3.5 RDBMS engine. If ARIES code were migrated unchanged to the new 4GL environment, the user could expect to see an approximate 20% increase in program execution time and a dramatic

increase in non-native data processing time. This would also greatly reduce the time required to build the data migration warehouse from USARC's FoxPro 2.0 data sources.

MapInfo and MapBasic have also been improved, however, the vendor has also created a separate integrated mapping component, called MapX, based on the Microsoft ActiveX standard. MapX allows developers to directly compile the desired mapping functionality with the prototyped application, thus eliminating the need for OLE communications between the two components. The integrated mapping features can now be "hardwired" directly into the ARIES U/I and would only need a fourth of the computer memory currently used by the MapInfo application.

b. ARIES DSS Component Redesign: The Need for Innovation

The current version of LDW in its 16-bit format, is still more than capable of providing the required functionality, but as USARC upgrades its IT infrastructure to take advantage of future technology, the ARIES DSS component in its unscalable form, will eventually fail to provide effective decision support. What is needed is a new scaleable and extensible 32-bit DSS component. Currently, there are no vendors marketing DSS software with all of the functionality of the 16-bit LDW application (e.g., automatic waiting, dynamic sensitivity, etc.) The difficult choice is to either wait for market conditions to change, or custom develop a similar DSS component with the following additional features:

- Allow the decision model to be directly manipulated within an integrated window on the ARIES U/I.
- Allow decision-makers to graphically add or delete new decision parameter nodes, as well as fully document how they are calculated.
- Create a Report Manager that allows the user to drag-and-drop various decision model graphs into standard documents (e.g., Microsoft Word, HTML, etc.)

c. Migrate Decision Making to the Web

In order to take advantage of USARC's upgraded Windows NT operating system and the continued focus on network-centric infrastructure; the competent developer can use the Visual Basic 5.0 component creation tool to build a new ARIES component that separates the USARC business rules and decision parameter calculations from the user interface. By using ActiveX technology, the ARIES SDSS can truly be a three-tier client-server application. The rules and processes could be changed, updated, or added separately from the ARIES U/I while still in accordance with corresponding changes in the ARU-Decision Model. Development time can now be better spent elsewhere instead of redesigning the U/I every time a business rule or condition changes.

The new Visual Basic development language also allows the developer to "web-enable" any application quickly and easily from within the 4GL environment. The ARIES U/I could now be rapidly prototyped within a common development environment by the collaborative team in less time, using a web browser (e.g., Netscape Navigator or Microsoft Explorer) and a point-and-click web page designer (e.g., Adobe PageMill or NetFusion). This would allow ARIES to use more of USARC's stabilized infrastructure, reduce the amount of software maintenance support needed, and reduce the number of languages and development tools that the application developer must learn.

2. Future Technology Options.

An important concern is how far the process of simplification of the user interface should go to provide easy access to spatial information without sacrificing necessary functionality. Prototyped software that enables site selection across space and time using a component-based infrastructure like the ARIES SDSS, has been created as one small step towards the continued development of spatial decision support systems. Thanks to

new component-based software tools, it is now possible to create compact, yet specialized web-based applications that allow inexperienced users to participate to a greater extent in solving local, spatially oriented problems as decision-makers rather than as programmers. This is all desirable as far as it goes, but without changes to other peripheral issues, the future capabilities of the ARIES NEXGEN can only advance so far. The USARC business processes related to site selection decisions are ripe for a concentrated reengineering effort that has the future in mind — Visioneering.

a. Data Source Management

The underlying data management philosophy of the NEXGEN SDSS is that the entire USARC network, local and remote sites, comprise a single, large virtual "Resources-to-Readiness" data migration warehouse. The decision-maker should be able to access any and every data source relating to site selection whether on an Army Reserve mainframe in Washington D.C., or a spatial data server on the local LAN; all from within the browser U/I. In today's network-centric ecosystem, it makes good sense to use the web as an operating desktop since it is a logical interface that most users can quickly learn and understand. Essentially, the NEXGEN is a network user interface (NUI), whereby the user is able to seamlessly apply business rules and calculate decision parameters without having to have knowledge of where the data actually resides.

The NUI will improve database processing by leveraging the flexibility of the web middleware to pull data simultaneously from all local and mainframe data stores using open standards. In short, web-based technologies act as a "data resolver" to connect decision-makers to information no matter where it is stored or how it is structured (Joch, 1997, p. 104D).

b. Decision-Maker Knowledge Management

In order to adapt to the continuous turnover of military personnel at USARC headquarters, the NEXGEN SDSS needs to be able to interactively model and capture the corporate knowledge about various data sources, business rules, and ad-hoc queries residing in the minds of the decision-makers. When stored and replicated, this meta-data knowledge becomes a reusable resource that ensures a given decision analysis can be based on repeatable findings. In short, this means knowledge management through automation.

c. Decision Parameter Derivation in Parallel

Although new technologies and software are produced everyday, the appetite for higher application performance is never satiated. The sequential calculation of ARIES decision parameters using spatial selection was a superior solution for mobile decision-makers, but within the NT ecosystem, parallelism is an even better solution. The NT network supports this by near simultaneous launching of remote automated OLE server objects for decision parameter computations onto the network. In this scenario, the idle processors of any desktop computers attached to the network can be utilized to calculate all the decision parameters essentially in parallel. Depending on the amount of network traffic present, the entire decision parameter matrix could be hypothetically calculated in the same amount of time it now takes to complete just one complex decision parameter query.

3. Future Scenario: The Final Episode.

> Users want to shape the technology that is implemented in their organization. They want to control its use and determine the effect it will have on their own work. They are rapidly understanding that their

> effective use of technology coupled with a change in how they do business will determine their personal and organizational success. (Tapscott, 1993, p. 14)

As component-based computing becomes more and more the norm for application development within civilian and military organizations, developing future IT decision support systems may become a far easier and user-friendlier process. Just ask Colonel McMurdo about his ARGIS 21 system:

1721 HRS: "Hey Jack! That ARGIS 21 product is your responsibility, right?" It was Brigadier General Mark Rutherford, Col. McMurdo's immediate boss, on the phone.

"Yes Sir," the Colonel replied.

"Well the Old Man just got a call about the BRAC situation on CNN. After that 'stick-save' at the hearing today, you're stock is up 20 prestige points. As a matter of fact, you have a meeting with the Chairman and the other Four Stars in the Tank first thing tomorrow morning Seems they have big plans for ARGIS 21 concerning better force dispersion, limited over-the-horizon target selection, and real-time Joint logistics shipping status. You'd better bring that laptop of yours."

"I'm on it, Sir," McMurdo responded, "and while I'm at it, I'd better give those people back at NPS a call. Looks like we're going to be quite busy."

APPENDIX A. ARIES SOURCE DATA FILE META-DATA.

This appendix contains the meta-data that was documented for the ARIES SDSS prototype data source files. "ACROPOLIS" as used in this appendix refers to the file name of the ARIES data migration warehouse (DMW).

Index

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: AMSA Location: ../Aries/MapBasic/USARCData

File Type: FoxPro 2.6 Size(MB): .026 No. Records: 190

Associated ARIES Tables: Not in ACROPOLIS, Geocoded for use in MapInfo

File Description:

AMSA File contains information about the location of each AMSA station. It is used in determining the value for the distance to the nearest AMSA.

Required Data Elements

Name	Description	Data Type	Format	Key Field
fac_id	Facility Identification Code	Char		No
fac_title	Facility Title	Char		No
fac_street	Street Address of Facility	Char		No
fac_city	City Facility is located in	Char		No
fac_state	State Facility is located in	Char		No
fac_zip	Zip Code of the Facility	Char		No
abb_type		Char		No

Extract Queries:

NONE

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: COMMAND PLAN Location: ACROPOLIS

File Type: FoxPro 2.6 Size(MB): 3.29 No. Records: 9,897

Associated ARIES Tables: CMDPLAN

File Description:

Command Plan is the file that contains information about each unit in the Army Reserve. It is used to cross reference FAC ID's with UIC's. It is also used to screen for Valid UIC's with in the next 13 months.

Required Data Elements

Name	Description	Data Type	Format	Key Field
UIC	Unit Identification Code	Char		yes
FACID	Facility Identification Code	Char		no
EDATE	Effective Date of Transaction	Char		no

Extract Queries:

```
CMDPLAN
SELECT DISTINCT UIC, FACID AS FAC_ID,
       EDATE
FROM   COMMANDPLAN
WHERE  (FACID <> "N/A") AND (FACID <>
       "TBD") AND (FACID <> "") AND
       (LEN(FACID) > 2) AND
       ((LEFT(EDATE,4) = '1998' AND
       MID(EDATE,5,2) <= '02') OR
       (LEFT(EDATE,4) <= '1997'))
ORDER BY  UIC, EDATE DESC
INTO   CMDPLAN
INDEX ON UIC as UIC
```

Note: Application automatically adjusts the dates to obtain a 13 month window.

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: COMPLEX Location: ACROPOLIS

File Type: FoxPro 2.6 Size(MB): 2.1 No. Records: 1,557

Associated ARIES Tables: COMPLEX_

File Description:

The Complex File is used to determine if the facility is owned by or leased to the government and the number of weekends each facility is used during a month.

Required Data Elements

Name	Description	Data Type	Format	Key Field
FAC_ID	Facility Identification Code	Char		yes
GOVT_OWN	Facility ownership status	Char	Y/N	no
RS_WKND_PM	Reserve Station weekend usage per mo.	Number	0-4	no

Extract Queries:

```
COMPLEX_
SELECT FAC_ID, GOVT_OWN AS
       FAC_OWNED, RS_WKND_PM AS
       FAC_WKND_USED
FROM   COMPLEX
WHERE  LEN(FAC_ID) = 5
INTO   COMPLEX_
INDEX ON FAC_ID as FACID, Primary, Unique
```

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: ECS Location: ...\MapBasic\USARCData\

File Type: FoxPro 2.6 Size(MB): .004 No. Records: 30

Associated ARIES Tables: Not in ACROPOLIS, Geocoded for use in MapInfo

File Description:

ECS File contains information about the location of each Equipment Center. It is used in determining the distance to the nearest ECS.

Required Data Elements

Name	Description	Data Type	Format	Key Field
fac_id	Facility Identification Code	Char		No
fac_title	Facility Title	Char		No
fac_street	Street Address of Facility	Char		No
fac_city	City Facility is located in	Char		No
fac_state	State Facility is located in	Char		No
fac_zip	Zip Code of the Facility	Char		No
abb_type		Char		No

Extract Queries:

NONE

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: FINANCE Location: ACROPOLIS

File Type: FoxPro 2.6 Size(MB): 83.4 No. Records: 311,793

Associated ARIES Tables: FINANCE_, FINANCE_QTR

File Description:

Finance is the file that contains pay information for the previous eight quarters about every Reservist. It is used to obtain information about Drill Attendance for a given Facility.

Required Data Elements

Name	Description	Data Type	Format	Key Field
CURR_UIC	Current Unit Identification Code	Char		No
UTA1QCFY	Unit Training Attendance for 1st Qtr this FY	Number		No
UTA2QCFY	Unit Training Attendance for 2nd Qtr this FY	Number		No
UTA3QCFY	Unit Training Attendance for 3rd Qtr this FY	Number		No
UTA4QCFY	Unit Training Attendance for 4th Qtr this FY	Number		No
UTA1Q1PF	Unit Training Attendance for 1st Qtr last FY	Number		No
UTA2Q1PF	Unit Training Attendance for 2nd Qtr last FY	Number		No
UTA3Q1PF	Unit Training Attendance for 3rd Qtr last FY	Number		No
UTA4Q1PF	Unit Training Attendance for 4th Qtr last FY	Number		No

Extract Queries:

FINANCE_

```
SELECT "W" & LEFT(CURR_UIC,5) AS UIC,
       COUNT(CURR_UIC) AS
       UIC_TOTAL
FROM   FINANCE
WHERE  CURR_UIC <> ""
ORDER BY  CURR_UIC
GROUP BY  CURR_UIC
INTO   FINANCE_
INDEX ON UIC as UIC
```

FINANCE_QTR

```
SELECT "W" & LEFT(CURR_UIC,5) AS UIC,
       UTA1QCFY, UTA2QCFY, UTA3QCFY,
       UTA4QCFY, UTA1Q1PF, UTA2Q1PF,
       UTA3Q1PF, UTA4Q1PF
FROM   FINANCE
WHERE  CURR_UIC <> "" AND NPS_IND = NULL
       AND PAY_STAT = 'A'
ORDER BY  CURR_UIC
INTO   FINANCE_QTR
INDEX ON UIC as UIC
```

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: FPS Location: ACROPOLIS

File Type: FoxPro 2.6 Size(MB): .088 No. Records: 1,561

Associated ARIES Tables: FPS_

File Description:

FPS is used to obtain information about the Cost to operate each facility as well as the Condition of each Facility. Used to return a value for the Cost per Square Foot and the Facility Condition.

Required Data Elements

Name	Description	Data Type	Format	Key Field
FAC_ID	Facility Identification Code	Char		No
FAC_COND	Condition of the Facility	Char		No
COST_PR_SF	Cost per Square Foot to Operate Facility	Number		No

Extract Queries:

```
FPS_
SELECT FAC_ID, FAC_COND, COST_PR_SF
FROM   FPS
WHERE  FAC_ID <> ""
ORDER BY  FAC_ID
INTO   FPS_
INDEX ON FAC_ID as FACID, Primary, Unique
```

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: FYxxLOSS Location: ACROPOLIS

File Type: FoxPro 2.6 Size(MB): 85.4 No. Records: 260,000

Associated ARIES Tables: FYxxLOSS, FYxxXFER

File Description:

FYxxLOSS file contains information about the personnel losses incurred by each unit during a fiscal year. It is used to determine the Average Loss and Transfer Rate of a Unit.

Required Data Elements

Name	Description	Data Type	Format	Key Field
UIC	Unit Identification Code	Char		No
TRMN	Transfer Reason Code	Char		No

Extract Queries:

FYxxLOSS

```
SELECT UIC1 AS UIC, COUNT(UIC1) AS
       UIC_TOTAL
FROM   FY_LOSS
WHERE  TRMN = 'LOSS'
ORDER BY UIC1
GROUP BY UIC1
INTO   FYxxLOSS
INDEX ON UIC as UIC, Primary, Unique
```

FYxxXFER

```
SELECT UIC1 AS UIC, COUNT(UIC1) AS
       UIC_TOTAL
FROM   FY_LOSS
WHERE  TRMN = 'TRFD'
ORDER BY   UIC1
GROUP BY   UIC1
INTO   FYxxXFER
INDEX ON UIC as UIC, Primary, Unique
```

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: G17 Location: ACROPOLIS

File Type: FoxPro 2.6 Size(MB): 3.11 No. Records: 5,869

Associated ARIES Tables: G17Natl

File Description:

G17 file contains facility Unitname, street address data and Zip Code. It is used as the primary cross reference with Command Plan to display facility information and validate user input.

Required Data Elements

Name	Description	Data Type	Format	Key Field
UIC	Unit Identification Code	Char		No
UNITNAME	Name of the Unit	Char		No
TCCCITY	City Unit is located in	Char		No
TCCSTATE	State Unit is located in	Char		No
TCCZIP	Zip code of the Unit	Char		No
TIER	Code used to determine if Unit is closing	Char		No
RECSTAT	Recruiting Station Code	Number		No
TYPEORG	Type of organization	Number		No

Extract Queries:

G17Natl

```
SELECT UIC, UNITNAME, TCCCITY AS
       CITY, TCCSTAT AS STATE,
       LEFT(TCCZIP,5) AS ZIP, TIER
FROM   G17
WHERE  (RECSTAT <> "1") AND (TYPEORG
       <> "2") AND UIC <> ""
ORDER BY  UIC
INTO   G17Natl
INDEX ON UIC as UIC, Primary, Unique
```

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: G18CWE Location: ACROPOLIS;..\MapBasic\UsarcData

File Type: FoxPro 2.6 Size(MB): 145.9 No. Records: 208,416

Associated ARIES Tables: G18Natl, G18Natl_UIC; also Geocoded for use in MapInfo

File Description:

G18 File contains information about personnel in the US Army Reserves. It is used in determining the Total Number Assigned used in calculating the Loss/Transfer Rates, Total Available Closing and the Reassignments values. Also used to obtain a list of the Zip Code's and MOSs of every Reservists with their associated UIC..

Required Data Elements

Name	Description	Data Type	Format	Key Field
UIC	Unit Identification Code assigned	Char		No
ZIP	Zip Code of the individual	Char		No
PRI	Primary MOS	Char		No

Extract Queries:

```
G18Natl
SELECT UIC, LEFT(ZIP,5) AS ZIPCODE,
       LEFT(PRI,3) AS MOS
FROM   G18_
WHERE  PRI <> "" AND UIC <> ""
ORDER BY UIC
INTO   G18Natl
INDEX ON UIC as UIC
```

```
G18Natl_UIC
SELECT  UIC, COUNT(UIC) AS UIC_TOTAL
FROM    G18Natl
ORDER BY   UIC
GROUP BY   UIC
INTO    G18Natl_UIC
 INDEX ON UIC as UIC, Primary, Unique
```

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: G19TRUE Location: ACROPOLIS

File Type: FoxPro 2.6 Size(MB): 14.4 No. Records: 233,211

Associated ARIES Tables: G19Natl

File Description:

G19 File contains information about the required manning levels of each Unit. It is used in determining Average Area Manning for a Facility.

Required Data Elements

Name	Description	Data Type	Format	Key Field
OWN_UIC	Unit Identification Code	Char		No

Extract Queries:

G19Natl

```
SELECT OWN_UIC AS UIC,
       COUNT(OWN_UIC) AS
       UIC_TOTAL
FROM   G19
WHERE  OWN_UIC <> ""
ORDER BY  OWN_UIC
GROUP BY  OWN_UIC
INTO   G19Natl
INDEX ON UIC as UIC
```

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: GEOREF Location: ACROPOLIS;..\MapBasic\UsarcData

File Type: FoxPro 2.6 Size(MB): .21 No. Records: 1,553

Associated ARIES Tables: VALID_UNIT; also Geocoded for use in MapInfo

File Description:

Georef File contains specific information about each Unit. It is used to verify and cross reference FACID's and UIC as well as Facility and Unit specific information.

Required Data Elements

Name	Description	Data Type	Format	Key Field
FAC_ID	Facility Identification Code	Char		No
FAC_TITLE	Name of the Facility	Char		No
FAC_CITY	City the Facility is located in	Char		No
FAC_STATE	State the Facility is located in	Char		No
FAC_ZIP	Zip Code of the Facility	Char		No
Latitude	Position of Facility by degree of latitude	Number		No
Longitude	Position of Facility by degree of longitude	Number		No

Extract Queries:

```
VALID_UNIT
SELECT FAC_ID, FAC_TITLE AS
        UNITNAME, FAC_CITY AS CITY,
        FAC_STATE AS STATE,
        LEFT(FAC_ZIP,5) AS ZIP
FROM    GEOREF
WHERE FAC_ID <> ""
ORDER BY  FAC_ID
INTO    VALID_UNIT
INDEX ON FAC_ID as FACID
```

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: INTEREST Location: ACROPOLIS

File Type: FoxPro 2.6 Size(MB): 4.2 No. Records: 3,985

Associated ARIES Tables: INTEREST_

File Description:

Interest File contains information about facilities and the date they were acquired. It is used to calculate the Facility Age for each facility..

Required Data Elements

Name	Description	Data Type	Format	Key Field
FAC_IDSTR	Facility Identification Code	Char		No
DATE_ACQ	Date Facility Acquired	Date		No
ABB_TYPE		Char		No

Extract Queries:

```
INTEREST_
SELECT FAC_IDSTR AS FAC_ID, DATE_ACQ
FROM   INTEREST
WHERE  FAC_IDSTR <> "" AND ABB_TYPE =
       "USARC (MB)"  AND NOT
       ISNULL(DATE_ACQ)
ORDER BY  FAC_IDSTR
INTO   INTEREST_
INDEX ON FAC_ID as FACID, Primary, Unique
```

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: IRR Location: ...\MapBasic\UsarcData

File Type: FoxPro 2.6 Size(MB): 7.5 No. Records: 140,077

Associated ARIES Tables: Not in ACROPOLIS, Geocoded for use in MapInfo

File Description:

IRR File contains information about the individuals listed in the Individual Ready Reserve. It is used to determine the value for IRR Available and Available MOS IRR.

Required Data Elements

Name	Description	Data Type	Format	Key Field
ZIPC	Zip Code for IRR Individual	Char		No

Extract Queries:

NONE

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: NGNON_CL Location: ...\MapBasic\UsarcData

File Type: FoxPro 2.6 Size(MB): .64 No. Records: 3,673

Associated ARIES Tables: Not in ACROPOLIS, Geocoded for use in MapInfo

File Description:

NGNON_CL File contains information about the non-closing National Guard Units. It is used in determining the value for Competition.

Required Data Elements

Name	Description	Data Type	Format	Key Field
UPC		Char		Yes
ZIP	Zip Code for National Guard Individual	Char		Yes

Extract Queries:

NONE

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: QMA Location: ...\MapBasic\UsarcData

File Type: FoxPro 2.6 Size(MB): 2.8 No. Records: 34,265

Associated ARIES Tables: Not in ACROPOLIS, Geocoded for use in MapInfo

File Description:

QMA File contains Census information. It is used in determining the value for Recruit Market for each Facility.

Required Data Elements

Name	Description	Data Type	Format	Key Field
ZIP	Zip Code	Char		No
MWCAT12	White Male Mental Categories 1 &2	Number		No
MWCAT3A	White Male Mental Category 3A	Number		No
MBCAT12	Black Male Mental Categories 1 &2	Number		No
MBCAT3A	Black Male Mental Category 3A	Number		No
MHCAT12	Hispanic Male Mental Categories 1 &2	Number		No
MHCAT3A	Hispanic Male Mental Category 3A	Number		No

Extract Queries:

NONE

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: RPINFODT Location: ACROPOLIS

File Type: FoxPro 2.6 Size(MB): 14.3 No. Records: 47,159

Associated ARIES Tables: FPINFODT_

File Description:

RPINFODT is a file that contains information about the backlogged maintenance costs of each Facility. It is used to determine the amount of backlogged maintenance is required at the given Facility.

Required Data Elements

Name	Description	Data Type	Format	Key Field
FAC_ID	Facility Identification	Char		No
CWE_TOTAL	Total amount of outstanding Maint. Actions	Number		No

Extract Queries:

```
RPINFODT_
SELECT FAC_ID, SUM(CWE_TOTAL) AS
       MAINT_COST
FROM   RPINFODT
WHERE  FAC_ID <> ""
ORDER BY  FAC_ID
GROUP BY  FAC_ID
INTO   RPINFODT_
INDEX ON FAC_ID as FACID, Primary, Unique
```

THIS PAGE LEFT INTENTIONALLY BLANK

A R I E S Data File Documentation Form

ARIES File Name: RZA Location: ...\MapBasic\UsarcData

File Type: FoxPro 2.6 Size(MB): .16 No. Records: 1,793

Associated ARIES Tables: Not in ACROPOLIS, Geocoded for use in MapInfo

File Description:

RZA File contains information about the location of Recruit Stations. It is used to determine the distance to the nearest Recruit Station.

Required Data Elements

Name	Description	Data Type	Format	Key Field
rsid	Recruit Station Identification Code	Char		No
name	Recruit Station Title	Char		No
zip	Zip Code of the Recruit Station	Char		No
latitude	Position of Recruit Station by latitude	Number		No
longitude	Position of Recruit Station by longitude	Number		No

Extract Queries:

NONE

THIS PAGE LEFT INTENTIONALLY BLANK

APPENDIX B. ARIES BUSINESS RULES AND PROCESSES.

This appendix contains detailed information about the calculation of each ARU-Decision Model parameter that was automated in the ARIES SDSS prototype application. The information includes a description of each decision parameter, the business rule used to calculate the associated value, base units, source files, associated data migration warehouse (ACROPOLIS) tables, query or queries involved in the calculation, a description of the yield curve, and a graph of the yield curves. "ACROPOLIS" is the file name for the ARIES data resource file.

The term "in the area" in this Appendix is defined as being with in a 50-mile radius of the moving unit or proposed facility.

Index

THIS PAGE LEFT INTENTIONALLY BLANK

Decision Parameter 1. Facility Backlogged Maintenance

Definition: Facility Backlogged Maintenance provides the total dollar value of backlogged maintenance. This provides an indication of the initial investment required to correct the significant maintenance problems with a proposed facility.

Calculation: The Backlogged Maintenance value is based upon the sum values for maintenance actions documented for each facility in the "CWE_TOTAL" field of the RPINFODT file. The summation is done during the data extraction phase.

Maint_Cost[Sum of outstanding maintenance actions for a facility]

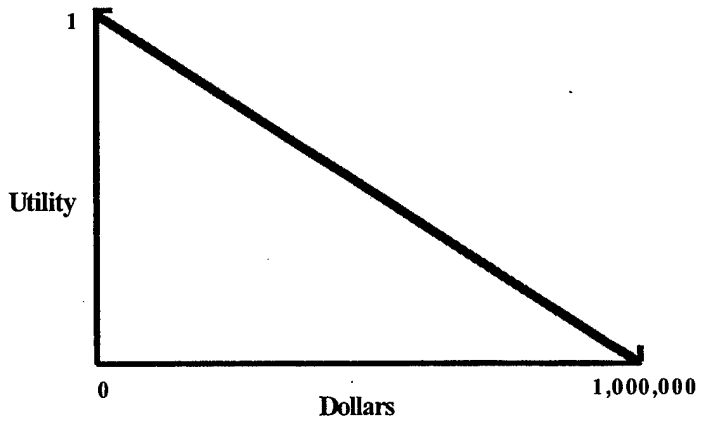
Units: Dollars

Source File: RPINFODT

ACROPOLIS Table(s): RPINFODT_

Query:
Maint_Cost
SELECT MAINT_COST
FROM RPINFODT_
WHERE RPINFODT_.FACID = ProposedFacility.FAC_ID

Yield Curve: A linear relationship is assumed between the backlogged maintenance costs and utility. Every dollar required or saved in this category is expected to have equal utility to a relocating unit.



Max Utility: 0 **Min Utility**: > 1,000,000

THIS PAGE INTENTIONALLY LEFT BLANK

DECISION PARAMETER 2. Facility Operating Costs

Definition: Facility Operating Costs provide an indication of the financial resources that are required to maintain the facility in a serviceable condition. This includes both utilities and minor maintenance costs.

Calculation: Operating Costs are extracted from the "COST_PR_SF" field of the FPS file.

COST_PR_SF[Retrieve the Cost per Square Foot for a facility]

Units: Dollars per square foot per month

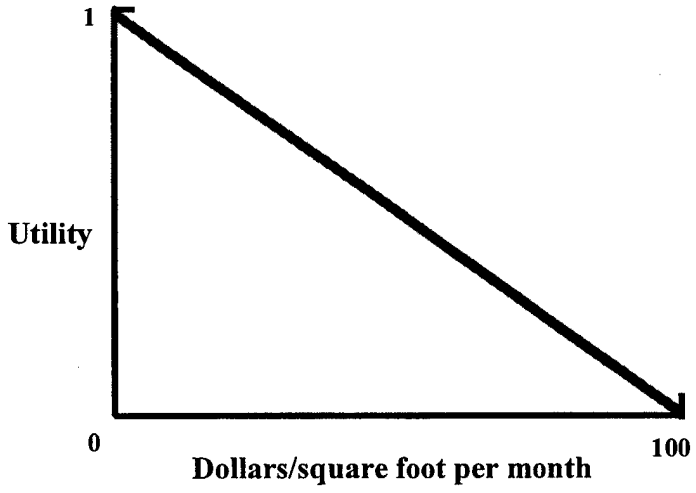
Source File: FPS

ACROPOLIS Table(s): FPS_

Query:

```
COST_PR_SF
SELECT  COST_PR_SF
FROM    FPS_
WHERE   FPS_.FACID = ProposedFacility.FAC_ID
```

Yield Curve: A linear relationship is assumed between the operating costs and utility. Every dollar required or saved in this category is expected to have equal utility to a relocating unit.



Max Utility: 0 **Min Utility**: > 100

THIS PAGE INTENTIONALLY LEFT BLANK

Decision Parameter 3. Facility Age

Definition: This Decision Parameter indicates the age of the primary structure on the proposed relocation site. It is intended to reflect an assumed long term structural degradation with time.

Calculation: Facility age is calculated based upon the acquisition date found in the INTEREST file. The acquisition date is compared to the current date and the difference is determined in months.

DATE_ACQ[Current Year - Date Acquired]

Units: Months

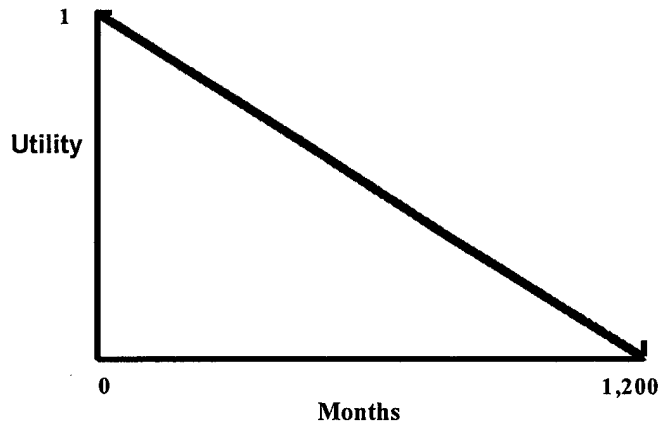
Source File: INTEREST

ACROPOLIS Table(s): INTEREST_

Query:

```
DATE_ACQ
SELECT  DATE_ACQ
FROM    INTEREST_
WHERE   INTEREST_.FACID = ProposedFacility.FAC_ID
```

Yield Curve: A linear relationship is used between facility age and utility.



Max Utility: 0 **Min Utility**: > 1,200

THIS PAGE INTENTIONALLY LEFT BLANK

Decision Parameter 4. Facility Condition

Definition: Facility Condition is based upon a visual inspection of the structure and provides an indication of the serviceability of the primary structures.

Calculation: This Decision Parameter is based upon the ISR part 1 rating entered in the "FAC_COND" field of the FPS file.

FAC_COND[Retrieve Facility Condition]

Units: No Units(Green, Amber, Red)

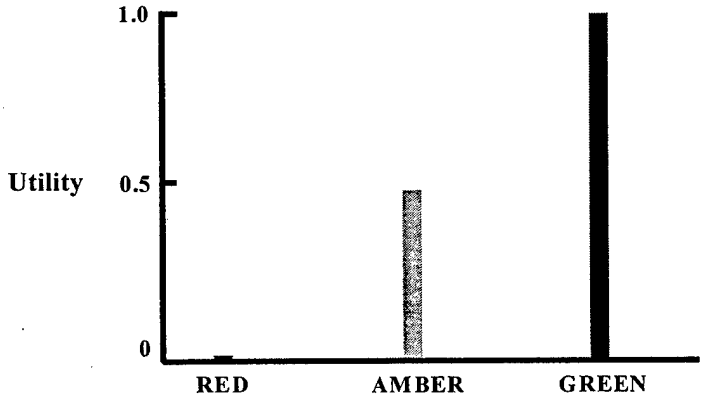
Source File: FPS

ACROPOLIS Table(s): FPS_

Query:

FAC_COND
SELECT FAC_COND
FROM FPS_
WHERE FPS_.FACID = ProposedFacility.FAC_ID

Yield Curve: The utility of these three categories varies in discrete steps. A facility that is categorized as "green" is judged to be approximately twice a desirable as one that is assigned an "amber" rating.



Max Utility: GREEN **Min Utility**: RED

THIS PAGE INTENTIONALLY LEFT BLANK

Decision Parameter 5. Facility Ownership

Definition: This Decision Parameter indicates whether the facilities at a proposed relocation site are leased or owned.

Calculation: Facility Ownership is based upon the entry in the "GOVT_OWN" field of the COMPLEX file.

GOVT_OWN[Retrieve Ownership Status]

Units: No Units(Yes/No)

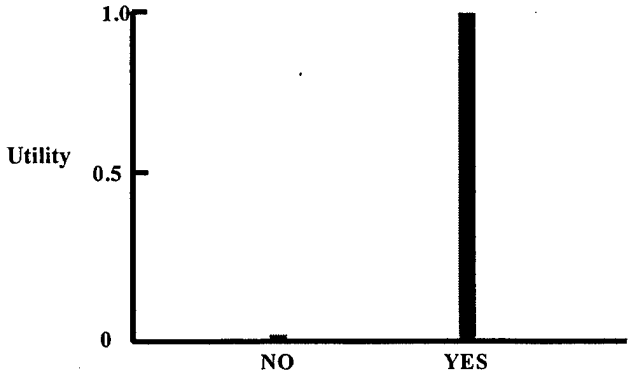
Source Data: COMPLEX

ACROPOLIS Table(s): COMPLEX_

Query:

GOVT_OWN
SELECT GOVT_OWN
FROM COMPLEX_
WHERE COMPLEX_.FACID = ProposedFacility.FAC_ID

Yield Curve: Facilities that are owned by the government are preferred as relocation sites over those facilities that are leased. The owned sites are assigned the maximum utility value of 1.0, while leased sites are given a 0 utility score.



Max Utility: Yes **Min Utility**: No

THIS PAGE INTENTIONALLY LEFT BLANK

Decision Parameter 6. Competition

Definition: This Decision Parameter provides an indication of the level of competition for potential reservists. It considers only Army Reserve and Army National Guard units in the area of the relocation site.

Calculation: Competition is determined by the number of positions that must be filled by all other Army Reserve and Army National Guard (ARNG) units in the area of the proposed relocation site. For Army Reserve units, the number of required positions is determined by counting the number of records in the G19TRUE file associated with each UIC in the area. For ARNG units, the value is found in the "AUTH" field of the NGNON_CL file.

NO_AUTH_NG[Number Authorized National Guard] +
NO_REQD[Number Area Reservists Required]

Units: Number of competing positions

Source File: COMMAND PLAN, G17, G19TRUE, GEOREF, NGNON_CL

ACROPOLIS Table(s): CMDPLAN, G17Natl, G19Natl, VALID_UIC

Query:

Area-FACID List(MapInfo)
SELECT FAC_ID INTO TempFACID
FROM GEOREF
WHERE Object Within ObjAreaBuffer
ORDER BY FAC_ID
(Note: ObjAreaBuffer is equal to 300 miles)

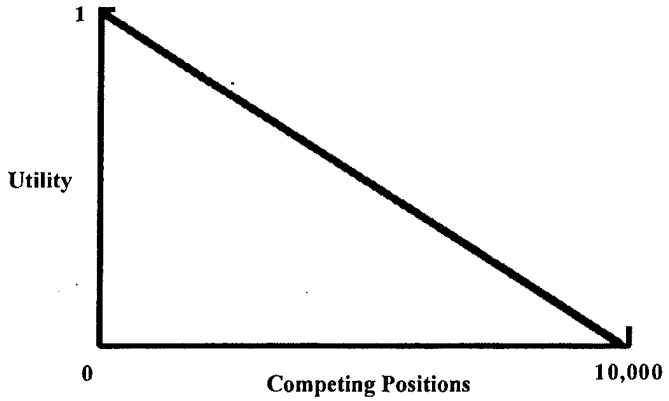
VALID_UIC
SELECT UIC, FAC_ID, UnitName, City, State, Zip
FROM G17Natl
WHERE G17Natl.UIC = ANY (SELECT CMDPLAN.UIC
FROM CMDPLAN)

Area-UIC List
SELECT DISTINCT UIC INTO AREA_UIC
FROM VALID_UIC
WHERE VALID_UIC.FAC_ID = ANY (SELECT AREA_FACID.FAC_ID
FROM AREA_FACID)

NO_AUTH_NG(MapInfo)
SELECT * INTO TempNGUnits
FROM NON_CLOS

```
WHERE   Obj Within ObjAreaBuffer

SELECT  SUM(AUTH) "No_AUTH_NG" INTO Strength
FROM    TempNGUnits

NO_REQD
SELECT  SUM(UIC_TOTAL) AS TOTAL_REQD
FROM    G19Natl
WHERE   G19Natl.UIC = ANY (SELECT AREA_UIC.UIC
                           FROM AREA_UIC)
```

Yield Curve: A linear relationship exists between the number of competing positions from other units and the utility of a relocation site. The level of no site utility in this Decision Parameter begins at 10,000 positions which is above the maximum value expected.



Max Utility: 0 **Min Utility**: > 10,000

Decision Parameter 7. Average Area Drill Attendance

Definition: This Decision Parameter indicates the fraction of reservists with satisfactory drill attendance for all existing units in the area of the proposed relocation site. Areas with a high fraction of satisfactory drill attendance are preferred relocation sites because units relocated to that area are assumed to perform similarly in drill attendance.

Calculation: This Decision Parameter considers the last four quarters of data contained in the FINANCE file. After initial screening, the number of reservist with 21 or more drill periods for the year is divided by the total number of people who meet the screening.

$$\frac{\text{DRILL\_SAT [Number of reservists with > 21 drill periods in a year]}}{\text{DRILL\_TOTAL [Number of reservists required to drill]}}$$

Units: Ratio

Source File: COMMAND PLAN, FINANCE, G17, G19TRUE, GEOREF

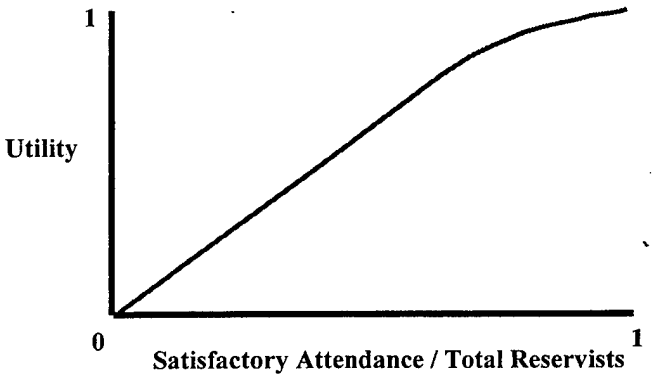
ACROPOLIS Table(s): CMDPLAN, FINANCE_, FINANCE_QTR, G17Natl, G19Natl, VALID_UIC

Query:

```
Area-FACID List(MapInfo)
SELECT   FAC_ID INTO TempFACID
FROM     GEOREF
WHERE    Object Within ObjAreaBuffer
ORDER BY  FAC_ID
(Note:  ObjAreaBuffer is equal to 300 miles)

VALID_UIC
SELECT   UIC, FAC_ID, UnitName, City, State, Zip
FROM     G17Natl
WHERE    G17Natl.UIC = ANY (SELECT CMDPLAN.UIC
                            FROM CMDPLAN)

Area-UIC List
SELECT   DISTINCT UIC INTO AREA_UIC
FROM     VALID_UIC
WHERE    VALID_UIC.FAC_ID = ANY (SELECT AREA_FACID.FAC_ID
                                 FROM AREA_FACID)
```

```
FINANCE_CY
SELECT   UIC, COUNT(UIC) AS UIC_TOTAL INTO FINANCE_CY
FROM     FINANCE_QTR
WHERE    (Select Case)
         Case 1st Qtr FY
         (UTA1Q1PF + UTA2Q1PF + UTA3Q1PF + UTA4Q1PF) > 20
         Case 2nd Qtr FY
         (UTA2Q1PF + UTA3Q1PF + UTA4Q1PF + UTA1QCFY) > 20
         Case 3rd Qtr FY
         (UTA3Q1PF + UTA4Q1PF + UTA1QCFY + UTA2QCFY) > 20
         Case 4th Qtr FY
         (UTA4Q1PF + UTA1QCFY + UTA2QCFY + UTA3QCFY) > 20
GROUP BY UIC
ORDER BY UIC

DRILL-SAT
SELECT   SUM(UIC_TOTAL) AS TOTAL_SAT
FROM     FINANCE_CY
WHERE    FINANCE_CY.UIC = ANY (SELECT AREA_UIC.UIC
                               FROM AREA_UIC)

DRILL-TOTAL
SELECT   SUM(UIC_TOTAL) AS DRILL_TOTAL
FROM     FINANCE_
WHERE    FINANCE_.UIC = ANY (SELECT AREA_UIC.UIC
                             FROM AREA_UIC.UIC)
```

Yield Curve: The utility of the average drill attendance rate increases linearly between the values of 0 and 0.6. Above that point, increases in the attendance rate result in diminishing returns. Values above 0.6 become increasingly uncommon.



Max Utility: 1.0 **Min Utility**: 0.0

Decision Parameter 8. Area Loss Rate

Definition: This Decision Parameter indicates the fraction of reservists who left the reserves in the previous fiscal year, for all existing units in the area of the proposed relocation site. Areas with a low loss rate are preferred relocation sites because units relocated to that area will also experience low loss rates.

Calculation: The number of losses to units in the area in the previous fiscal year is divided by the number of reservists currently assigned to these units. Losses are identified through the transfer mnemonic field (TRMN="LOSS") of the FyxxLOSS file. The number of assigned reservists is determined by counting all of the personnel records in the G18CWE file associated with each UIC in the area.

$$\frac{\text{NO\_LOSS[Total Number of Losses in the last year]}}{\text{NO\_ASSN[Total Number Reservists Assigned]}}$$

Units: Ratio

Source File: COMMAND PLAN, FYxxLOSS, G17, G18CWE, GEOREF

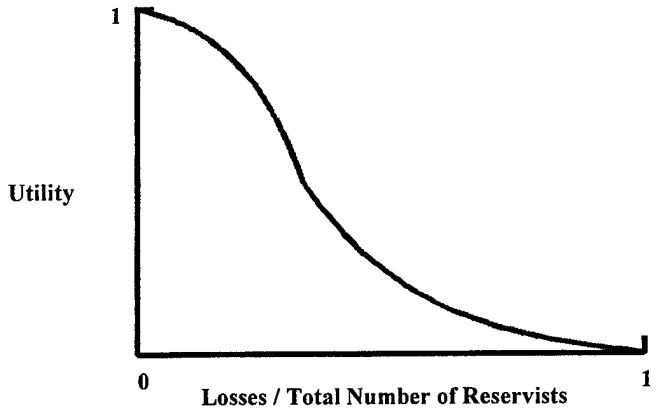
ACROPOLIS Table(s): CMDPLAN, FYxxLOSS, G17Natl, G18Natl_UIC, VALID_UIC

Query:

```
Area-FACID List(MapInfo)
SELECT   FAC_ID INTO TempFACID
FROM     GEOREF
WHERE    Object Within ObjAreaBuffer
ORDER BY  FAC_ID
(Note:  ObjAreaBuffer is equal to 300 miles)

VALID_UIC
SELECT   UIC, FAC_ID, UnitName, City, State, Zip
FROM     G17Natl
WHERE    G17Natl.UIC = ANY (SELECT CMDPLAN.UIC
                            FROM CMDPLAN)

Area-UIC List
SELECT   DISTINCT UIC INTO AREA_UIC
FROM     VALID_UIC
WHERE    VALID_UIC.FAC_ID = ANY (SELECT AREA_FACID.FAC_ID
                                 FROM AREA_FACID)

NO_ASSN
SELECT   SUM(UIC_TOTAL) AS TOTAL_ASSN
FROM     G18Natl_UIC
```

```
WHERE   G18Natl_UIC.UIC = ANY (SELECT AREA_UIC.UIC
                                FROM AREA_UIC)

NO_LOSS
SELECT  SUM(UIC_TOTAL) AS TOTAL_LOSS
FROM    FYxxLOSS
WHERE   FYxxLOSS.UIC = ANY (SELECT AREA_UIC.UIC
                             FROM AREA_UIC)
```

Yield Curve: This function includes both concave and convex regions. The inflection point occurs at a loss rate of .33 and a utility of 0.5. Based on experience, a loss rate of one third per year was considered to be typical. Any loss rate below this value has relatively high utility, whereas loss rates above the inflection point quickly approach a utility of zero.



Max Utility: 0 **Min Utility**: 1

Decision Parameter 9. Area Transfer Rate

Definition: This Decision Parameter indicates the fraction of reservists who transferred to different units in the previous fiscal year for all existing units in the area of the proposed relocation site. Areas with a low transfer rate are preferred relocation sites because units relocated to that area will also experience low transfer rates.

Calculation: The number of transfers in the previous fiscal year is divided by the number of reservists currently assigned to the unit. Transfers are identified through the transfer mnemonic field (TRMN="TRFD") of the FyxxLOSS file. The number of assigned reservists is determined by counting all of the personnel records in the G18CWE file associated with each UIC.

$$\frac{\text{NO\_XFER[Total Number of Transfers in the last year]}}{\text{NO\_ASSN[Total Number Reservists Assigned]}}$$

Units: Ratio

Source File: COMMAND PLAN, FYxxLOSS, G17, G18CWE, GEOREF

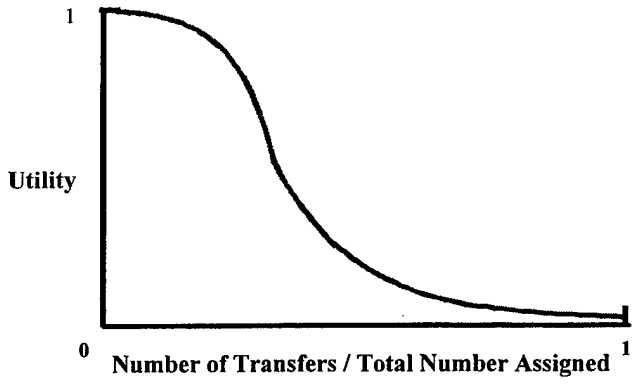
ACROPOLIS Table(s): CMDPLAN, G17Natl, G18Natl_UIC, FYxxXFER, VALID_UIC

Query:

```
Area-FACID List(MapInfo)
SELECT   FAC_ID INTO TempFACID
FROM     GEOREF
WHERE    Object Within ObjAreaBuffer
ORDER BY  FAC_ID
(Note:  ObjAreaBuffer is equal to 300 miles)

VALID_UIC
SELECT   UIC, FAC_ID, UnitName, City, State, Zip
FROM     G17Natl
WHERE    G17Natl.UIC = ANY (SELECT CMDPLAN.UIC
                            FROM CMDPLAN)

Area-UIC List
SELECT   DISTINCT UIC INTO AREA_UIC
FROM     VALID_UIC
WHERE    VALID_UIC.FAC_ID = ANY (SELECT AREA_FACID.FAC_ID
                                 FROM AREA_FACID)
```

```
NO_ASSN
SELECT   SUM(UIC_TOTAL) AS TOTAL_ASSN
FROM     G18Natl_UIC
WHERE    G18Natl_UIC.UIC = ANY (SELECT AREA_UIC.UIC
                                FROM AREA_UIC)

NO_XFER
SELECT   SUM(UIC_TOTAL) AS TOTAL_XFER
FROM     FYxxXFER
WHERE    FYxxXFER.UIC = ANY (SELECT AREA_UIC.UIC
                             FROM AREA_UIC)
```

Yield Curve: This function includes both concave and convex regions. The inflection point occurs at a loss rate of .33 and a utility of 0.5. Based on experience, a transfer rate of one third per year was considered to be typical. Any loss rate below this value has relatively high utility (close to 1.0), whereas loss rates above the inflection point quickly approach a utility of zero.



Max Utility: 0 **Min Utility**: 1

Decision Parameter 10. Area Average Manning

Definition: This Decision Parameter indicates the ability to fill the required positions. An average value is determined for all existing units in the area of the proposed relocation site. Areas with high average manning levels are preferred relocation sites because units relocated to that area will also experience high manning levels.

Calculation: The number of reservists assigned to area units (based upon the number of personnel records in G18CWE file associated with each UIC) is divided by the number of required positions (based upon the number of positions in the G19TRUE file associated with each UIC). An average is calculated for all UIC's in the area of the proposed site.

$$\frac{\text{NO\_ASSN[Total Number Reservists Assigned]}}{\text{NO\_REQD[Number Area Reservists Required]}}$$

Units: Ratio

Source File: COMMAND PLAN, G17, G18CWE, G19TRUE, GEOREF

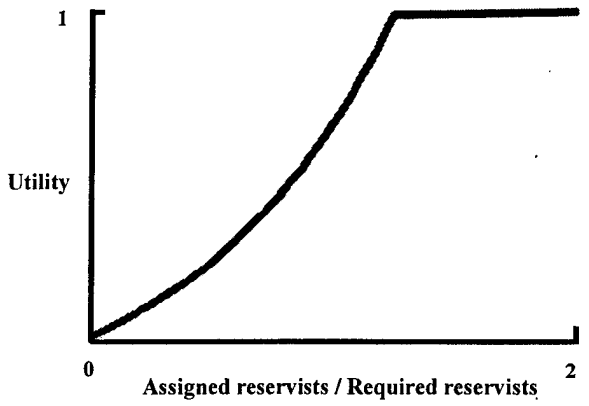
ACROPOLIS Table(s): CMDPLAN, G17Natl, G18Natl_UIC, G19Natl, VALID_UIC

Query:

```
Area-FACID List(MapInfo)
SELECT   FAC_ID INTO TempFACID
FROM     GEOREF
WHERE    Object Within ObjAreaBuffer
ORDER BY  FAC_ID
(Note:  ObjAreaBuffer is equal to 300 miles)

VALID_UIC
SELECT   UIC, FAC_ID, UnitName, City, State, Zip
FROM     G17Natl
WHERE    G17Natl.UIC = ANY (SELECT CMDPLAN.UIC
                            FROM CMDPLAN)

Area-UIC List
SELECT   DISTINCT UIC INTO AREA_UIC
FROM     VALID_UIC
WHERE    VALID_UIC.FAC_ID = ANY (SELECT AREA_FACID.FAC_ID
                                 FROM AREA_FACID)
```

```
NO_ASSN
SELECT   SUM(UIC_TOTAL) AS TOTAL_ASSN
FROM     G18Natl_UIC
WHERE    G18Natl_UIC.UIC = ANY (SELECT AREA_UIC.UIC
                                FROM AREA_UIC)

NO_REQD
SELECT   SUM(UIC_TOTAL) AS TOTAL_REQD
FROM     G19Natl
WHERE    G19Natl.UIC = ANY (SELECT AREA_UIC.UIC
                            FROM AREA_UIC)
```

Yield Curve: It is desirable that area units be able to exceed their minimum manning requirements. All manning levels above 125% are considered to have maximum utility. Manning levels below this value drop off quickly in terms of utility.



Max Utility: 1.25 **Min Utility**: 0

Decision Parameter 11. Distance to Nearest Recruit Station

Definition: Distance to the nearest Recruiting Station provides one indication of recruiter effectiveness.

Calculation: The straight-line distance from the proposed site to the closest recruiting station is calculated using a geocoded version of the RZA file.

DIST_RZA[Determine distance to nearest Recruit Station]

Units: Miles

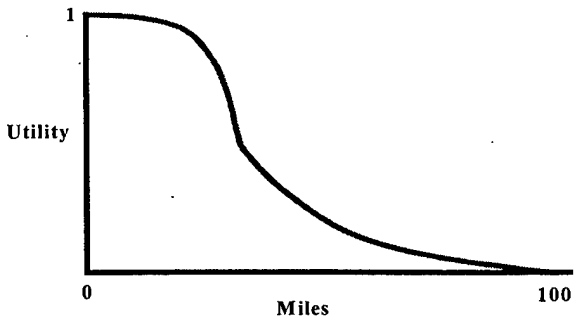
Source Data: RZA

ACROPOLIS Table(s): NONE

Query:

```
DIST_RZA(MapInfo)
SELECT  *
FROM    RZA
WHERE   Obj Withing ObjDistanceBuffer into TempRZA
(Note:  ObjDistanceBuffer is equal to 300 miles)

SELECT  Distance((CentroidX(Obj), CentroidY(Obj), FacIDLat, FacIDLong, "mi")
FROM    TempRZA
ORDER BY  Distance INTO TempRZA.Dist
```

Yield Curve: The effectiveness of a recruiting station in filling positions at a reserve unit is fairly high if the two are within a half hour drive of each other. It is assumed that recruiters are most effective in the area close to their recruiting station and that reserve recruits must be located near the unit with which they will serve. A distance of 30 miles is assigned an average utility of 0.5. A small change in distance results in less change in desirability when the distance is very small or very large than it does when the distance is around 30 miles.



Max Utility: 0 **Min Utility**: > 100

THIS PAGE INTENTIONALLY LEFT BLANK

Decision Parameter 12. Available Transfers from Closing Units

Definition: This value indicates the total number of personnel assigned to closing units within 50 miles of the proposed site.

Calculation: A list of Unit Identification Codes (UIC's) is created which contains only those units scheduled to close within 18 months. These units are identified by an entry of 5B in the "Tier" field of the G17 file. The number of potential transfers from closing units is calculated by summing the number of records in the G18CWE database for the closing units which are located in the area of the proposed relocation site.

TOTAL_AVAIL[Total Number of Available Reservists from Area Closing Units]

Units: Ratio

Source File: COMMAND PLAN, G17, G18CWE, GEOREF, US_ZIPS(MapInfo)

ACROPOLIS Table(s): CMDPLAN, G17Natl, VALID_UIC

Query:

```
Area-FACID List(MapInfo)
SELECT   FAC_ID INTO TempFACID
FROM     GEOREF
WHERE    Obj Within objAreaBuffer
ORDER BY  FAC_ID
(Note:  objAreaBuffer is equal to 300 miles)

VALID_UIC
SELECT   UIC, FAC_ID, UnitName, City, State, Zip
FROM     G17Natl
WHERE    G17Natl.UIC = ANY (SELECT CMDPLAN.UIC
                            FROM CMDPLAN)

Area-UIC List
SELECT   DISTINCT UIC INTO AREA_UIC
FROM     VALID_UIC
WHERE    VALID_UIC.FAC_ID = ANY (SELECT AREA_FACID.FAC_ID
                                 FROM AREA_FACID)

AREA_CLOS_UIC
SELECT   UIC
FROM     G17Natl
WHERE    G17Natl.TIER = "5B"
         AND G17Natl.UIC = ANY (SELECT AREA_UIC.UIC
```

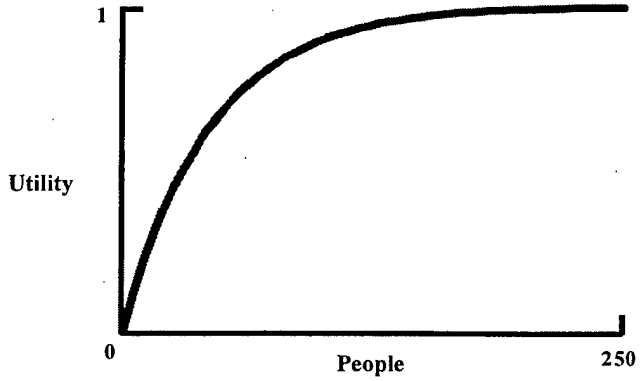
FROM AREA_UIC)

AREA_ZIPCODE(MapInfo)
SELECT ZIP_CODE
FROM US_ZIPS
WHERE Obj Within objAreaBuffer
ORDER BY ZIP_CODE

Area_G18_ZIP(MapInfo)
SELECT DISTINCT UIC, ZIPCODE, COUNT(UIC) AS UIC_TOTAL
FROM G18CWE
GROUP BY UIC, ZIPCODE
ORDER BY UIC, ZIPCODE

TOTAL_AVAIL
SELECT SUM(UIC_TOTAL) AS TOTAL_AVAIL
FROM Area_G18_ZIP
WHERE Area_G18_ZIP.UIC = ANY (SELECT AREA_CLOS_UIC.UIC
FROM AREA_CLOS_UIC)
AND Area_G18_ZIP.ZIPCODE = ANY (SELECT AREA_ZIPCODE.ZIP
FROM AREA_ZIPCODE)

Yield Curve: The shape of this function assumes diminishing returns in the number of transfers available. Experience suggests that for an average unit of 100 people, approximately half have prior reserve experience and that approximately half of the people in a closing unit will be able to transfer their skills directly to a new unit. The value of the first 100 reservists increases at a nearly linear rate because they provide preferred fills for approximately 50 of the positions of the moving unit. A value of 100 personnel is assigned a utility of 0.9. The incremental value added by each additional person over 100 continues to drop until no marginal gain is expected over 500.



Max Utility: > 250 **Min Utility**: 0

Decision Parameter 13. IRR Available

Definition: Individual Ready Reserve (IRR) Available is the number of IRR members living in the area of the proposed relocation site. This is a Decision Parameter of the size of the prior service market.

Calculation: A geographical query returns the total number of IRR members living within a specified distance of the proposed relocation site. This process requires a geocoded version of the IRR file.

TOTAL_IRR[Total Number of Available IRR from the Area]

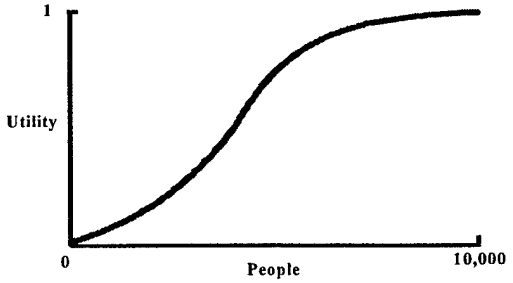
Units: People

Source File: IRR

Query:

```
AreaIRR(MapInfo)
SELECT   ZIPC "ZIP", LEFT$(PMOS, 3) "MOS"
FROM     IRR
WHERE    Obj Within objAreaBuffer
         AND ZIPC <> "" AND PMOS <> ""
ORDER BY ZIPC
(Note:  objAreaBuffer is equal to 300 miles)

TOTAL_IRR
SELECT   COUNT(*) AS TOTAL_IRR
FROM     AreaIRR
```

Yield Curve: For a typical unit of 100 people, it is assumed that approximately 40 positions could best be filled by IRR members. The recruiting rate for the IRR is approximately 1 percent, so an area that offers 4,000 IRR members is assigned an average utility of 0.5. Above this point, there are diminishing returns. The market begins to exceed the personnel demand of a moving unit and limited recruiting efforts become marginally less effective. The utility of smaller numbers quickly drops off because of the importance of this source of recruits.



Max Utility: > 10,000 **Min Utility**: 0

THIS PAGE INTENTIONALLY LEFT BLANK

Decision Parameter 14. Recruit Market

Definition: The Recruit Market Decision Parameter estimates the total number of males who:
1. live in the area of the proposed relocation site
2. Would score in the top half on the Armed Forces Qualification Test (AFQT)
3. Fall into the desired age group (17 - 29 years old)

Calculation: This Decision Parameter sums the entries for all mental categories 1 through 3A, and all ethnic groups for the zip codes of interest in the Qualified Military Available (QMA) file. The version of QMA used contains only the estimates for males within the age range of 17 to 29.

TOTAL_MARKET[Total Non-Prior Service Personnel from the Area]

Units: People

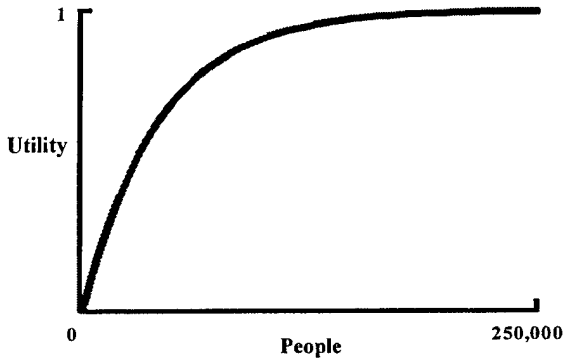
Source File: QMA, US_ZIPS(MapInfo)

ACROPOLIS Table(s): NONE

Query:

```
QMA(MapInfo)
SELECT  LEFT$(ZIP, 5) "ZIPCODE", MWCAT12, MWCAT3A, MBCAT12,
        MBCAT3A, MHCAT12, MHCAT3A
FROM    QMA
WHERE   Obj Within objAreaBuffer
ORDER BY ZIP
(Note:  objAreaBuffer is equal to 300 miles)

AREA_ZIPCODE(MapInfo)
SELECT  ZIP_CODE
FROM    US_ZIPS
WHERE   Obj Within objAreaBuffer
ORDER BY   ZIP_CODE

TOTAL_MARKET
SELECT  SUM(MWCAT12+MWCAT3A+MBCAT12+MBCAT3A+
        MHCAT12+MHCAT3A) AS TOTAL_MARKET
FROM    QMA
WHERE   QMA.ZIP = ANY (SELCET AREA_ZIPCODE.ZIP
                       FROM AREA_ZIPCODE)
```

Yield Curve: Approximately half of a typical unit of 100 reservists is filled by recruits with no prior service. Assuming a recruit rate of 0.25 percent, there must be at least 20,000 people in the area of the proposed relocation site who meet all of the requirements stated above. This value is assigned a typical utility of 0.5. As the number increases, there are diminishing returns. The market begins to exceed the personnel demand of a moving unit and limited recruiting efforts become marginally less effective.



Max Utility: > 250,000 **Min Utility**: 0

Decision Parameter 15. Reassignments

Definition: The Reassignments Decision Parameter indicates the fraction of the reservists assigned to the moving unit who currently live within a specified distance (50 miles) of the proposed relocation site

Calculation: This Decision Parameter is calculated by first determining all zip codes that lie within a specified distance of the proposed relocation site (based upon zip code centroid) and then identifying all reservists who both live within one of the identified zip codes (based upon the "ZIP" field of the G18CWE file) and are assigned to the moving unit (based upon the "UIC" field of the G18CWE file). Then the number available reassignments is divided by the total number of reservists assigned to the moving unit.

$$\frac{\text{TOTAL\_RESERVISTS[Total Number of Available Reservists from the Moving Unit]}}{\text{UIC\_TOTAL[Total Number of Reservists Assigned Moving Unit]}}$$

Units: Ratio

Source File: G18CWE, US_ZIPS(MapInfo)

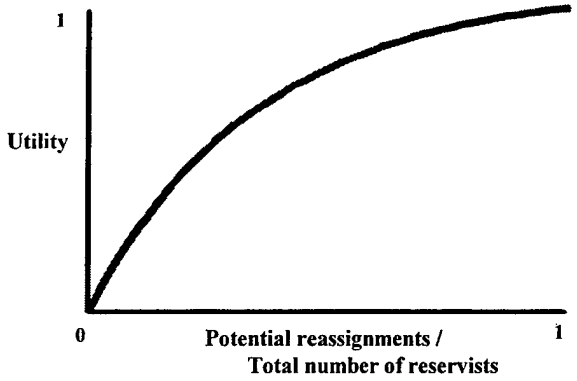
ACROPOLIS Table(s): G18Natl

Query:

```
AREA_ZIPCODE(MapInfo)
SELECT   ZIP_CODE
FROM     US_ZIPS
WHERE    Obj Within objAreaBuffer
ORDER BY   ZIP_CODE

G18(MapInfo)
SELECT   UIC, LEFT$(ZIP,5) "ZIPCODE", PRI "MOS"
FROM     G18CWE
WHERE    Obj Within objG18Buffer AND PRI <> ""
ORDER BY   UIC, ZIP
INTO     G18

Area_G18_ZIP
SELECT   DISTINCT UIC, ZIPCODE, COUNT(UIC) AS UIC_TOTAL
FROM     G18
GROUP BY   UIC, ZIPCODE
ORDER BY   UIC, ZIPCODE
```

```
TOTAL_RESERVISTS
SELECT   SUM(UIC_TOTAL) AS TOTAL_RESERVISTS
FROM     Area_G18_ZIP
WHERE    Area_G18_ZIP.UIC = MovingUnit.UIC
         AND Area_G18_ZIP.ZIPCODE = ANY (SELECT AREA_ZIPCODE.ZIP
                                         FROM AREA_ZIPCODE)

UIC_TOTAL
SELECT   UIC_TOTAL
FROM     G18Natl
WHERE    G18Natl.UIC = MovingUnit.UIC
```

Yield Curve: The current location will always receive a utility score of 0.0 on this Decision Parameter. For relatively close relocation sites, this function was made to be convex, assigning high utility values to alternatives that are close to the current location.



Max Utility: 1.0 **Min Utility**: 0.0

Decision Parameter 16. Distance to Area Maintenance Support Activity

Definition: Distance to the nearest Area Maintenance Support Activity (AMSA) is calculated as a proxy Decision Parameter for response time and support quality.

Calculation: The straight-line distance from the proposed site to the closest AMSA is calculated using a geocoded version of the AMSA file.

DIST_AMSA[Determine distance to nearest AMSA Site]

Units: Miles

Source Data: AMSA

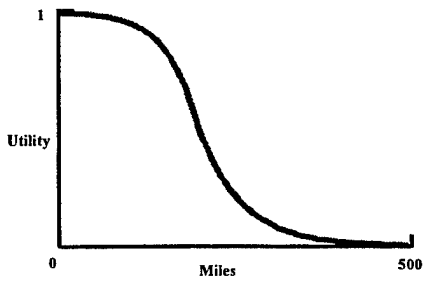
ACROPOLIS Table(s): NONE

Query:

DIST_AMSA(MapInfo)
SELECT *
FROM AMSA
WHERE Obj Withing ObjDistanceBuffer into TempRZA
(Note: ObjDistanceBuffer is equal to 300 miles)

SELECT Distance((CentroidX(Obj), CentroidY(Obj), FacIDLat, FacIDLong, "mi")
FROM TempAMSA
ORDER BY Distance INTO TempAMSA.Dist

Yield Curve: The desirability of a relocation site is relatively insensitive to small changes in distance for both close and distant AMSA sites. Little degradation in service is expected if the AMSA can have parts and technicians on site within a couple hours using a car or truck. It is possible that a trainer that breaks down in the morning may be operational for an afternoon training session. At approximately 200 miles (assigned a 0.5 utility) it starts to become impractical to expect same day service and avoid an overnight stay. Eventually it becomes necessary to consider flying rather than driving which is likely to further reduce the responsiveness and effectiveness of the AMSA.



Max Utility: 0 **Min Utility**: > 500

THIS PAGE INTENTIONALLY LEFT BLANK

Decision Parameter 17. Distance to Nearest Equipment Concentration Site

Definition: Distance to the nearest Equipment Concentration Site (ECS) provides an indication of the training time that must be used to travel back and forth.

Calculation: The straight-line distance from the proposed site to the closest ECS is calculated using a geocoded version of the ECS file.

DIST_ECS[Determine distance to nearest ECS]

Units: Miles

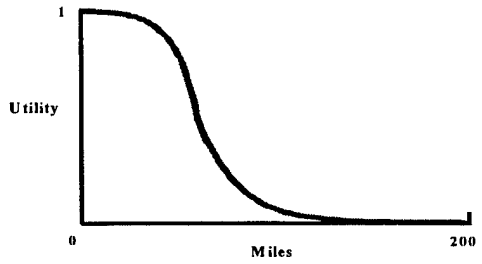
Source Data: ECS

ACROPOLIS Table(s): NONE

Query:

```
DIST_ECS(MapInfo)
SELECT  *
FROM    ECS
WHERE   Obj Withing ObjDistanceBuffer into TempECS
(Note:  ObjDistanceBuffer is equal to 300 miles)

SELECT  Distance((CentroidX(Obj), CentroidY(Obj), FacIDLat, FacIDLong, "mi")
FROM    TempECS
ORDER BY  Distance INTO TempECS.Dist
```

Yield Curve: The desirability of an Equipment Concentration Site is relatively insensitive to small changes in distance for both close and distant sites. Typically, a site that can be reached within an hour and ten minutes is not significantly less desirable than one that can be reached in ten minutes. An hour of one-way travel time is not normally considered to be excessive and allows for most of the time to be spent training on a one day training exercise. At approximately 60 miles (assigned a 0.5 utility) it starts to become impractical to expect useful training to be conducted on a day trip and avoid an overnight stay. Eventually it becomes necessary to consider flying rather than driving which is likely to further reduce the desirability of the ECS.



Max Utility: 0 **Min Utility:** > 200

THIS PAGE INTENTIONALLY LEFT BLANK

Decision Parameter 18. Facility Weekends Used

Definition: Facility Weekend Usage provides the number of weekends per month that the facility is currently in use. This Decision Parameter treats a facility as a limited resource that is incrementally depleted as more units are assigned. Since most units require exclusive use of the facility one weekend every month, the number of weekends used normally corresponds to the number of units assigned and is typically limited to four.

Calculation: This value is extracted from the "RS_WKND_PM" field of the COMPLEX file.

WKND_USED[Retrieve Number Weekends Facility Used per Month]

Units: Weekends per month

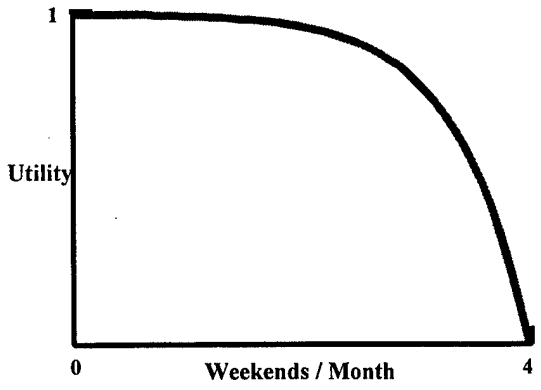
Source Data: COMPLEX

ACROPOLIS Table(s): COMPLEX_

Query:

```
WKND_USED
SELECT  COMPLEX_.FAC_WKND_USED
FROM    COMPLEX_
WHERE   COMPLEX_.FAC_ID = ProposedFacility.FAC_ID
```

Yield Curve: Although some exceptions exist, a typical facility offers no utility to a relocating unit if all four weekends are already being used. Although most facilities with three units or less should be able to accommodate a new unit and might be viewed as having equal utility, other issues such as full time administrative space and available equipment storage space make a facility with fewer units currently assigned slightly more desirable.



Max Utility: 0 **Min Utility**: 4

THIS PAGE INTENTIONALLY LEFT BLANK

Decision Parameter 19. Available MOS from Closing Units

Definition: This Decision Parameter provides the number of reservists from closing units in the area of the proposed relocation site who possess a Military Occupational Specialty (MOS) needed by the relocating unit. These people provide a preferred pool of trained and qualified recruits.

Calculation: The number of personnel records (from the G18CWE file) that meet all the following requirements are counted:

1. The reservist is assigned to a unit that is scheduled to close (a TIER="5B" entry in the G17 file is used to produce a list of closing units).
2. The reservist lives in a zip code in the area of the proposed relocation site.
3. The reservist's primary MOS is needed by the moving unit.

If the three MOS groups with the largest number of members in the moving unit account for more than 50 percent of the total unit membership, then only those three MOS's are considered. Otherwise all MOS's required by the moving unit are considered as an MOS of interest.

TOTAL_CLOS_MOS[Total Number of Available Reservists from Area Closing Units with MOS's of Interest]

Units: Number of people

Source File: COMMAND PLAND, G17, G18CWE, GEOREF, US_ZIPS(MapInfo)

ACROPOLIS Table(s): CMDPLAN, G17Natl, G18Natl, VALID_UIC

Query:

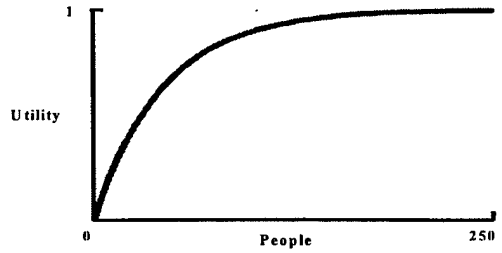
Area-FACID List(MapInfo)
SELECT FAC_ID INTO TempFACID
FROM GEOREF
WHERE Obj Within objAreaBuffer
ORDER BY FAC_ID
(Note: objAreaBuffer is equal to 300 miles)

VALID_UIC
SELECT UIC, FAC_ID, UnitName, City, State, Zip
FROM G17Natl
WHERE G17Natl.UIC = ANY (SELECT CMD_PLAN.UIC
FROM CMD_PLAN)

```
Area-UIC List
SELECT   DISTINCT UIC INTO AREA_UIC
FROM     VALID_UIC
WHERE    VALID_UIC.FAC_ID = ANY (SELECT AREA_FACID.FAC_ID
                                 FROM AREA_FACID)

NoAssnxMOS
SELECT   MOS, COUNT(*) AS MOS_COUNT INTO NoAssnxMOS
FROM     G18Natl
WHERE    G18Natl.UIC = MovingUnit.UIC
GROUP BY  MOS
ORDER BY  COUNT(*) DESC

MOS_TOTAL
SELECT   SUM(MOS_COUNT) AS MOS_TOTAL
FROM     NoAssnxMOS

MOS_TOP3
SELECT   TOP 3 MOS_COUNT
FROM     NoAssnxMOS

MOS_INTEREST
IF MOS_TOP3/MOS_TOTAL < 50%
   SELECT    MOS INTO MOS_INTEREST
   FROM      NoAssnxMOS
   ORDER BY  MOS
IF MOS_TOP3/MOS_TOTAL > 50%
   SELECT    TOP 3 MOS INTO MOS_INTEREST
   FROM      NoAssnxMOS
   ORDER BY  MOS

AREA_CLOS_UIC
SELECT   UIC
FROM     G17Natl
WHERE    G17Natl.TIER = "5B"
         AND G17Natl.UIC = ANY (SELECT AREA_UIC.UIC
                                FROM AREA_UIC)

AREA_ZIPCODE(MapInfo)
SELECT   ZIP_CODE AS ZIP
FROM     US_ZIPS
WHERE    Obj Within objAreaBuffer
ORDER BY   ZIP_CODE
```

```
G18(MapInfo)
SELECT   UIC, LEFT$(ZIP,5) "ZIPCODE", PRI "MOS"
FROM     G18CWE
WHERE    Obj Within objG18Buffer AND PRI <> ""
ORDER BY   UIC, ZIP
INTO     G18

Area_G18_MOS
SELECT   DISTINCT UIC, ZipCode, MOS, COUNT(UIC) AS UIC_TOTAL
INTO     Area_G18_MOS
FROM     G18
GROUP BY  UIC, ZipCode, MOS
ORDER BY  UIC, ZipCode, MOS

Area_G18_ZIP
SELECT   DISTINCT UIC, ZIPCODE, COUNT(UIC) AS UIC_TOTAL
FROM     G18
GROUP BY   UIC, ZIPCODE
ORDER BY   UIC, ZIPCODE

TOTAL_CLOS_MOS
SELECT   SUM(UIC_TOTAL) AS TOTAL_CLOS_MOS
FROM     Area_G18_MOS
WHERE    Area_G18_MOS.MOS = ANY (SELECT MOS_INTEREST.MOS
                                 FROM MOS_INTEREST)
         AND Area_G18_ZIP.UIC = ANY (SELECT AREA_CLOS_UIC.UIC
                                     FROM AREA_CLOS_UIC)
         AND Area_G18_ZIP.ZIPCODE = ANY (SELECT AREA_ZIPCODE.ZIP
                                         FROM AREA_ZIPCODE)
```

Yield Curve: The shape of this function assumes diminishing returns on the number of transfers available. Experience suggests for an average unit of 100 people, that it is unusual to expect more than a third of the members to transfer from closing units with the proper MOS. Of the reservists in this category, only half typically transfer, so a value of 60 personnel is assigned a utility of 0.9. The incremental value added by each additional person over 60 continues to drop until no marginal gain is expected over 250.



Max Utility: > 250 **Min Utility**: 0

THIS PAGE INTENTIONALLY LEFT BLANK

Decision Parameter 20. Available MOS IRR

Definition: This Decision Parameter provides the number of Individual Ready Reserve members who live in the area of the proposed relocation site and who possess a Military Occupational Specialty (MOS) needed by the relocating unit. These people provide a preferred pool of trained recruits.

Calculation: The number of IRR members who possess an MOS needed by the moving unit and who live in the area of the proposed relocation site (based upon the zip code of their home of record in the IRR file) are counted. If the three MOS groups with the largest number of members in the moving unit account for more than 50 percent of the total unit membership, then only those three MOSs are considered. Otherwise all MOSs required by the moving unit are considered as an MOS of interest.

TOTAL_IRR_MOS[Total Number of Available Reservists from the IRR with MOS's of Interest]

Units: Number of People

Source File: IRR, G18CWE

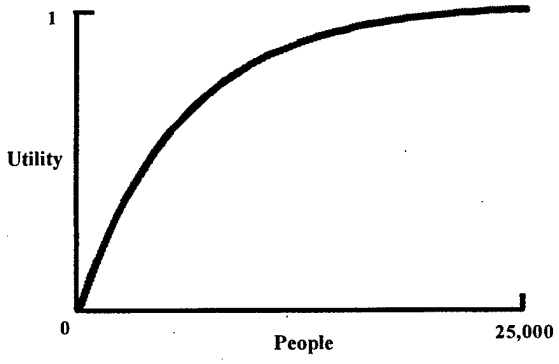
ACROPOLIS Table(s): G18Natl,

Query:

```
NoAssnxMOS
SELECT   MOS, COUNT(*) AS MOS_COUNT INTO NoAssnxMOS
FROM     G18Natl
WHERE    G18Natl.UIC = MovingUnit.UIC
GROUP BY  MOS
ORDER BY  COUNT(*) DESC

MOS_TOTAL
SELECT   SUM(MOS_COUNT) AS MOS_TOTAL
FROM     NoAssnxMOS

MOS_TOP3
SELECT   TOP 3 MOS_COUNT
FROM     NoAssnxMOS

MOS_INTEREST
IF MOS_TOP3/MOS_TOTAL < 50%
    SELECT    MOS INTO MOS_INTEREST
    FROM      NoAssnxMOS
    ORDER BY  MOS
```

```
IF MOS_TOP3/MOS_TOTAL > 50%
    SELECT    TOP 3 MOS INTO MOS_INTEREST
    FROM      NoAssnxMOS
    ORDER BY  MOS

IRR(MapInfo)
SELECT   ZIPC "ZIP", LEFT$(PMOS, 3) "MOS"
FROM     IRR
WHERE    Obj Within objAreaBuffer and ZIPC <> "" AND PMOS <> ""
ORDER BY ZIPC
INTO     IRR

TOTAL_IRR_MOS
SELECT   SUM(UIC_TOTAL) AS TOTAL_CLOS_MOS
FROM     IRR
WHERE    IRR.MOS = ANY (SELECT MOS_INTEREST.MOS
                        FROM MOS_INTEREST)
```

Yield Curve: IRR members represent preferred recruits for less than half of the positions of a typical moving unit (approximately 40 out of 100) because of issues such as seniority and changes in the skills associated with an MOS. The success rate of recruiting IRR members is approximately 1 out of 100, so 4000 IRR members in the area of the relocation site are required to provide sufficient market to fill the 40 positions. The value of 4000 is assigned the average utility value of 0.5. As the IRR market increases it exceeds the needs of the moving unit and makes the limited recruiting efforts marginally less effective.



Max Utility: > 25,000 **Min Utility**: 0

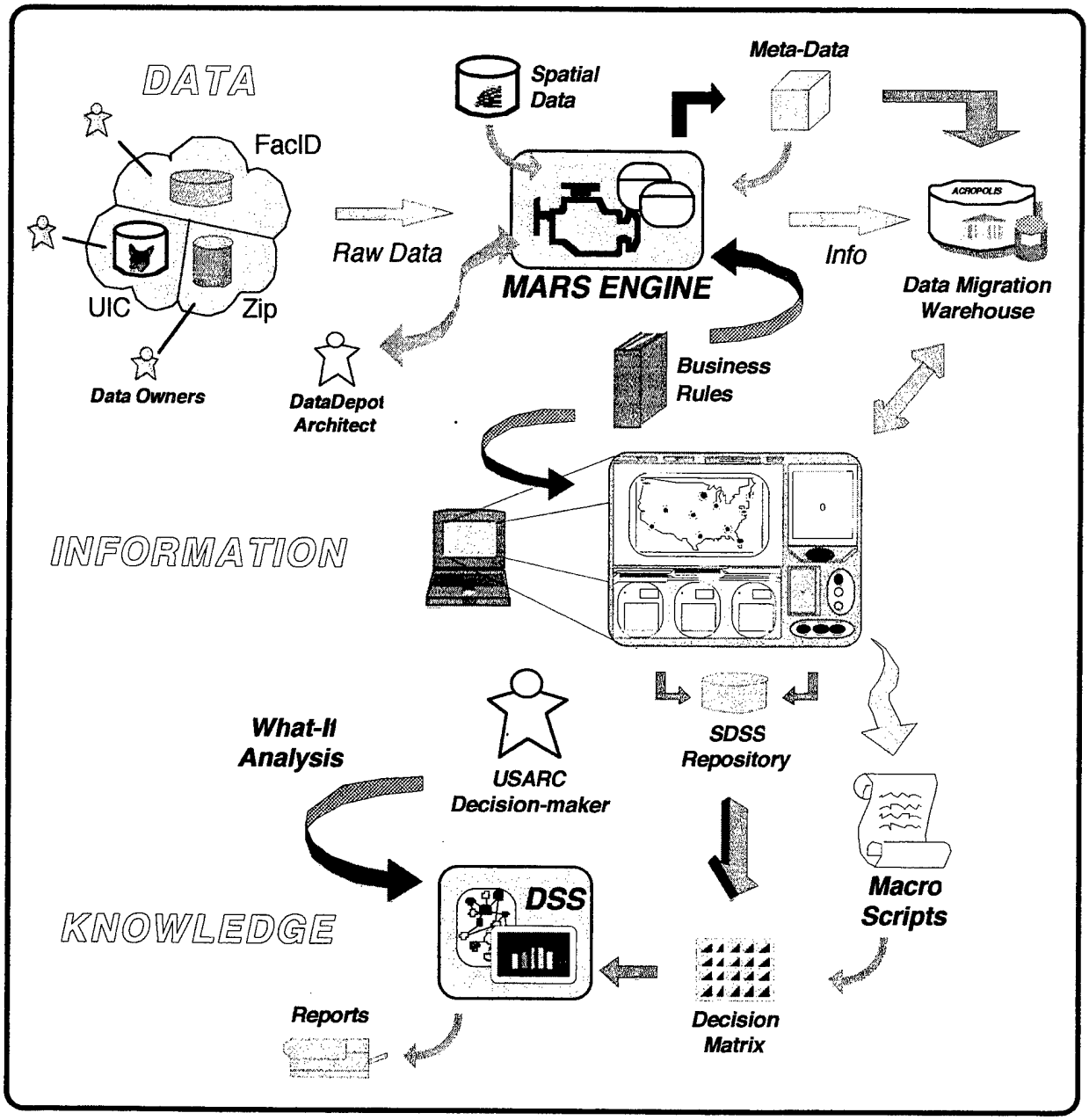
APPENDIX C. CONCEPTUALIZATION PHASE ARTIFACTS.

This appendix contains the artifacts that were documented upon completion of the Concept-to-Code (C2C) Conceptualization Phase.

Index

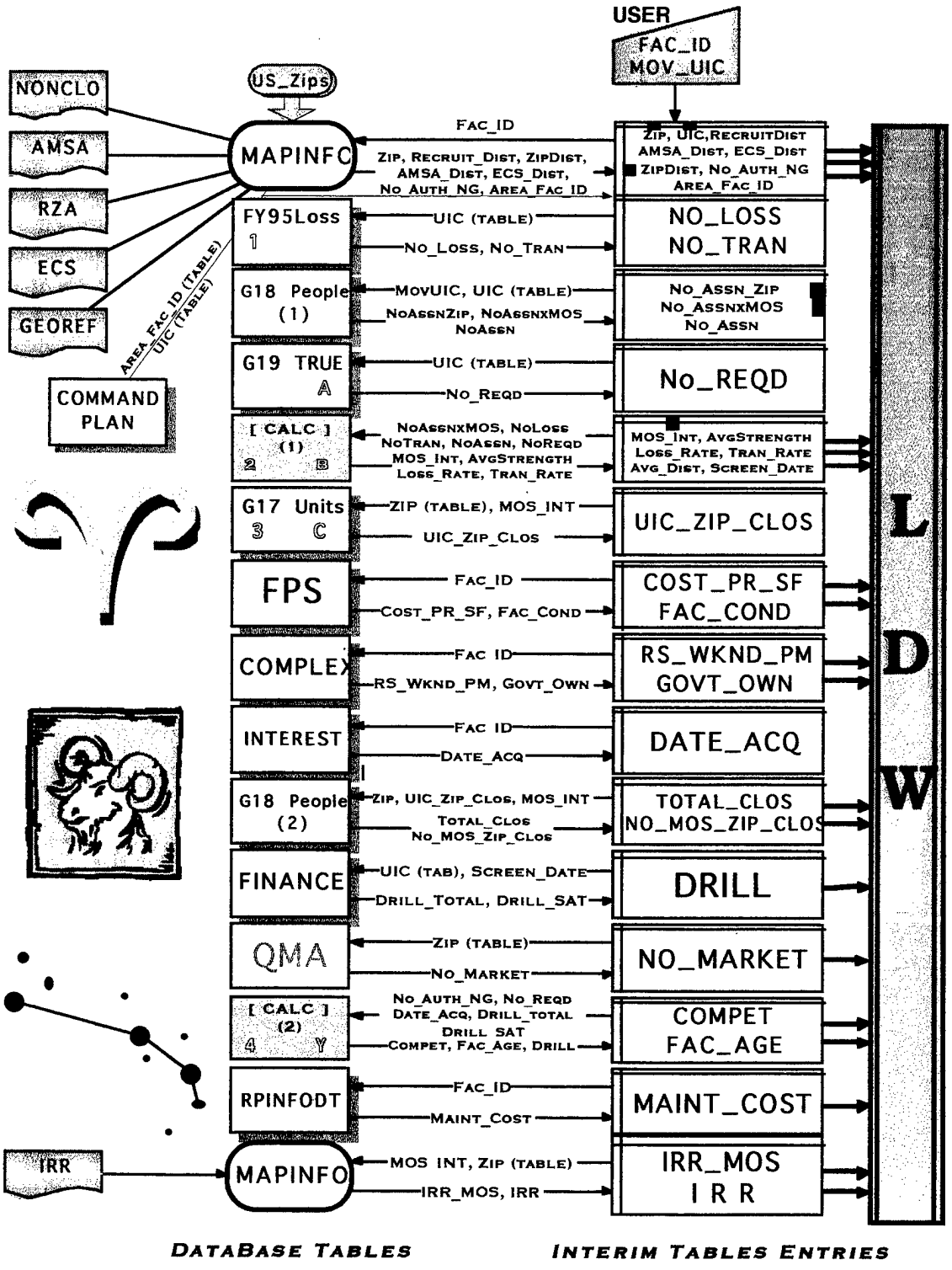
THIS PAGE LEFT INTENTIONALLY BLANK

Conceptual Overview Diagram.



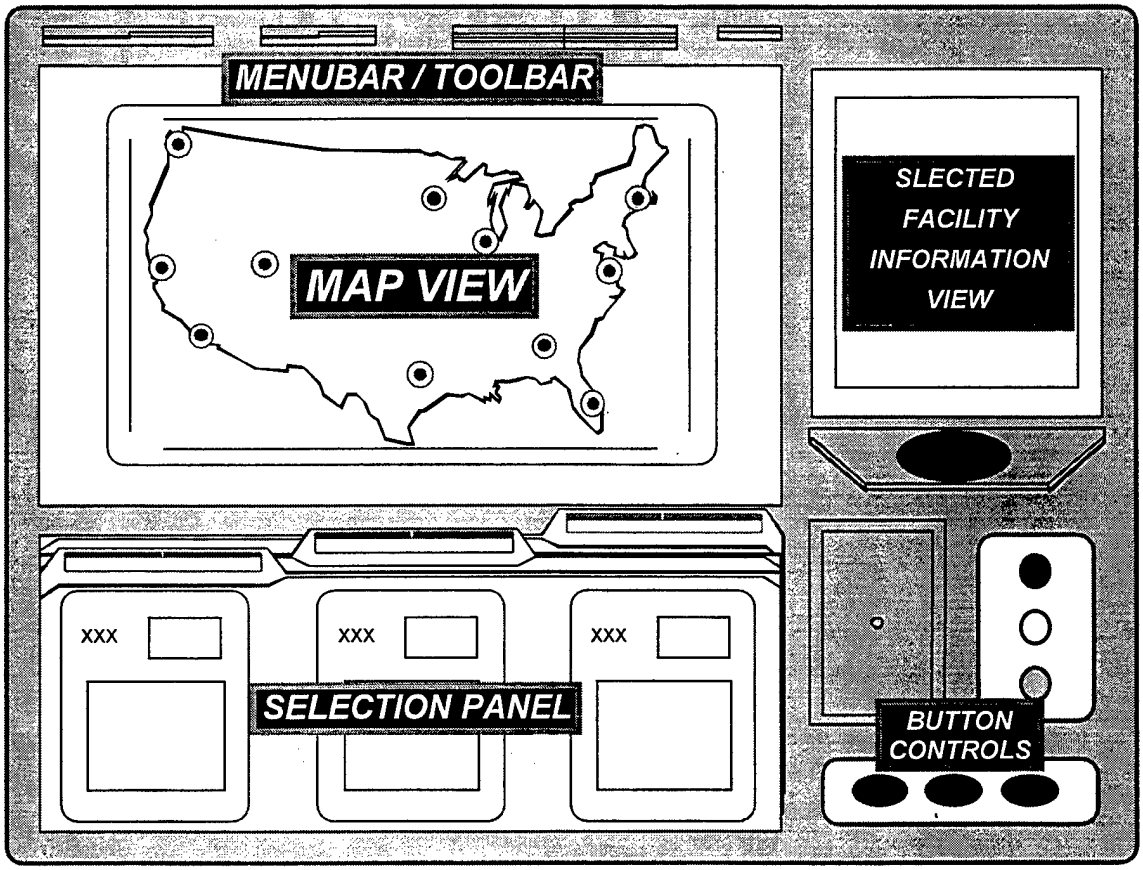
THIS PAGE LEFT INTENTIONALLY BLANK

ARIES Process Flow Diagram.



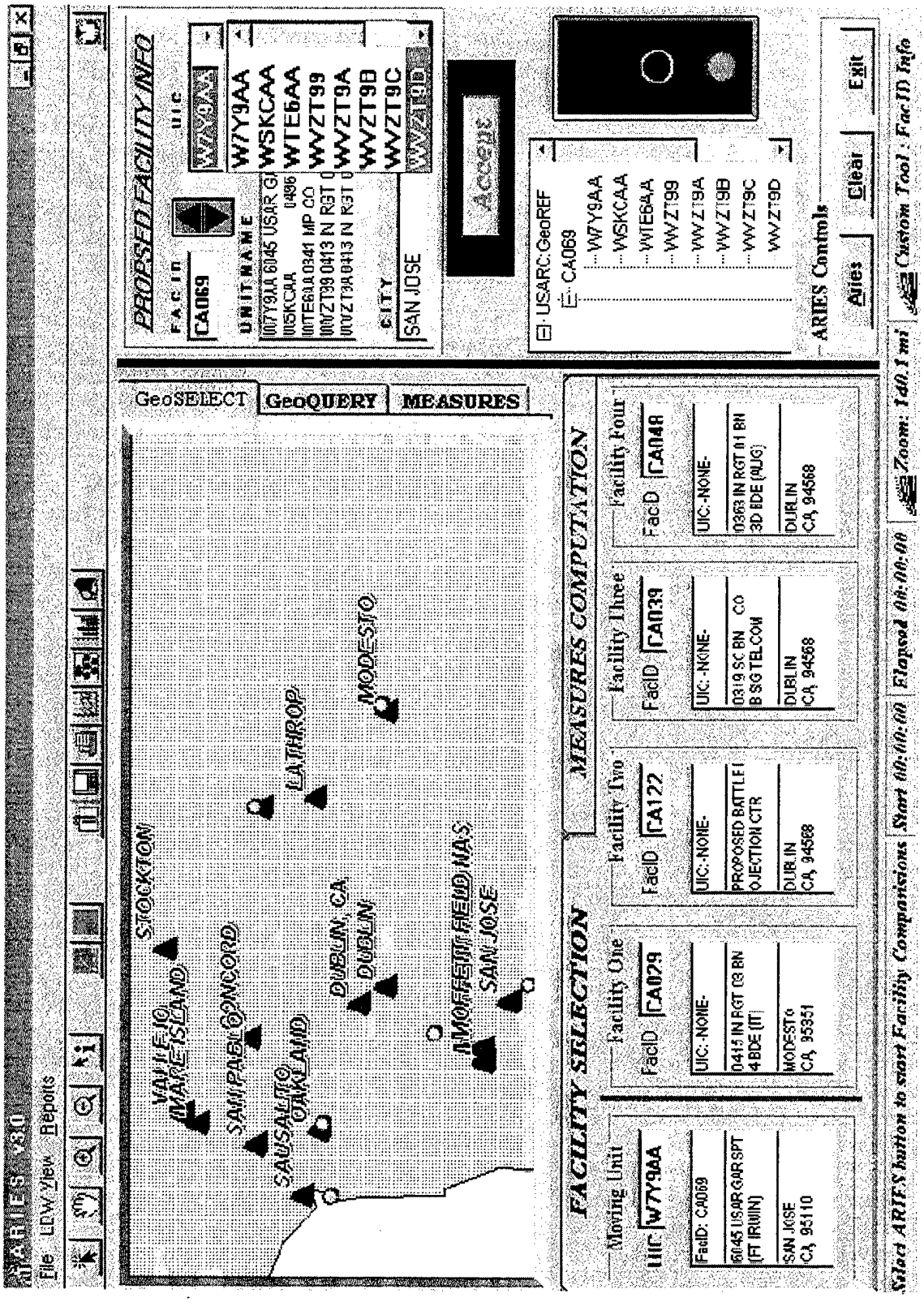
THIS PAGE LEFT INTENTIONALLY BLANK

User Interface Interim Layout.



THIS PAGE LEFT INTENTIONALLY BLANK

User Interface Final Layout.



APPENDIX D. ARIES SDSS CODE LISTINGS.

This appendix contains the detailed code listings of each Visual and Map Basic module that comprises the ARIES SDSS prototype application.

Index

THIS PAGE LEFT INTENTIONALLY BLANK

Module 1. SUBMAIN PROCEDURE

Purpose: Submain Procedure immediately executes upon ARIES application loading in memory. It is a standard Visual Basic initialization procedure. For the purposes of the ARIES SDSS, the procedure loads the splash screen for 30 seconds, initializes all OLE connections with MapInfo components and displays the main SDSS User Interface form.

Source Type: Visual Basic for Applications

Source File: SUBMAIN.BAS

Code Listing:

```
Attribute VB_Name = "SubMain"

Option Explicit

'Allocate Module Variables
Public bValidUIC As Boolean
Public bSQLQueryDone As Boolean
Public bLDWactive As Boolean
Public bGEOREF_Exists As Boolean

Public bMapQueryActive As Boolean
Public bMapInfoRunning As Boolean
Public bMapBasicRunning As Boolean

Public iPropFacIDctr As Integer
Public iProgressIndicator As Integer
Public iStatusBarMax As Integer

'WinAPI32 window handles
Public iLDWwinID As Long
Public iMapInfoWinID As Long

Public aFacID(4) As String * 5

Public cQueryStartTime As String * 8
Public cProcName As String

'Allocates Aries Measures Table
Public cMovUic As MovingUnit_Type

'Allocate Database Objects
Public dbAcropolis As Database
Public dbTeliko As Database
Public dbWorkspc As Workspace

Public objExcel As Object
Public objGeoSelect As Object
Public objGeoQuery As Object
Public objCallback As Object


Public Sub Main()
    cProcName = "SUB_MAIN"

On Error GoTo EH_SubMain
```

```
    Set objExcel = Nothing
    Set objGeoSelect = Nothing
    Set objGeoQuery = Nothing
    Set objCallback = Nothing

'Initialize all Boolean FLAGS to Default value.
   bValidUIC = False
   bLDWactive = False                    'LDW Activation Flag
   bSQLQueryDone = False
   bGEOREF_Exists = True

   bMapQueryActive = False               'MapBasic Activation Flag
   bMapBasicRunning = False
   bMapInfoRunning = False

'Initialize API hWnd ID variables
   iLDWwinID = 0
   iMapInfoWinID = 0

'Commence loading ARIES Application
   Load frmSplashScrn

'bLDWactive = OpenLDWobject
'Call ImportLDWdata
'Call ViewLDWmatrix

Exit Sub

EH_SubMain:     'Submain Error Handler

   Select Case Err.Number

      Case ERR_ObjectUnloaded
         Call DisplayMessage(OLEfailedMSG, vbExclamation, vbOKOnly,
    conOLEcaption)
         End

      Case Else    'Trap all other Errors & Report
         Call OutputERROR_LOG
         Resume Next

   End Select
End Sub

Public Sub DisplayUserInterface()
   cProcName = "Display_User_Interface"

   With frmAriesMain
      .Visible = True
      .WindowState = vbMaximized
      .Show
   End With
   Screen.MousePointer = vbDefault
End Sub
```

Module 2. SDSS USER INTERFACE

Purpose: Contains all procedures, business rules, and logic for controlling the ARIES U/I

Source Type: Visual Basic for Applications

Source File: USERINTERFACE.FRM

Code Listing:

```
Attribute VB_Name = "frmAriesMain"
Attribute VB_Creatable = False
Attribute VB_Exposed = False

Option Explicit

Private Sub mnuFOpen_Click()
   Dim iLDW_hWnd As Long

   On Error GoTo EH_FileOpen

   dlgLDWFiler.Flags = cdlOFNPathMustExist + cdlOFNFileMustExist
   dlgLDWFiler.InitDir = App.Path & FILE_LDW_ArchiveDir
   dlgLDWFiler.filename = vbNullString
   dlgLDWFiler.ShowOpen

   iLDW_hWnd = FindWindow(Class_LDW, vbNullString)
   If iLDW_hWnd = 0 Then Call OpenLDWobject

   ShowWindow iLDW_hWnd, SW_Minimum
   ShowWindow iLDW_hWnd, SW_Normal

   SendKeys "%FN", True
   SendKeys "%FO", True
   SendKeys dlgLDWFiler.filename, True
   SendKeys "{ENTER}", True
   DoEvents
   SendKeys "%VM~", True

Exit Sub

EH_FileOpen:

   If Not Err.Number = cdlCancel Then Call OutputERROR_LOG

End Sub

Private Sub mnuFSave_Click()
   Dim iLDW_hWnd As Long

   On Error GoTo EH_FileSave

   dlgLDWFiler.Flags = cdlOFNPathMustExist
   dlgLDWFiler.InitDir = App.Path & FILE_LDW_ArchiveDir
   dlgLDWFiler.filename = vbNullString
   dlgLDWFiler.ShowSave

   iLDW_hWnd = FindWindow(Class_LDW, vbNullString)
   If iLDW_hWnd = 0 Then Call OpenLDWobject

   ShowWindow iLDW_hWnd, SW_Minimum
   ShowWindow iLDW_hWnd, SW_Normal
```

```
    SendKeys "%FA", True
    SendKeys dlgLDWFiler.filename, True
    SendKeys "{ENTER}", True
    DoEvents

    AppActivate ARIES_Application
Exit Sub

EH_FileSave:

    If Not Err.Number = cdlCancel Then Call OutputERROR_LOG

End Sub

Private Sub mnuFCloseLDW_Click()

    If VerifyLDWobjectConnection(LDWactiveMSG) Then
       SendKeys "%FX", True
       DoEvents
       mnuVImport.Enabled = True
       tbararies.Buttons("import").Enabled = True
    End If
End Sub

Private Sub mnuFCloseMap_Click()
    Dim iMapInfo_hWnd As Long

    iMapInfo_hWnd = FindWindow(Class_MapInfo, vbNullString)
    If iMapInfo_hWnd = 0 Then
       Call DisplayMessage(APIfailedMSG, vbCritical, vbOKOnly, conAPIcaption)
    Else
       bMapInfoRunning = False
       ShowWindow iMapInfo_hWnd, SW_Minimum
       ShowWindow iMapInfo_hWnd, SW_Normal

       SendKeys "%FX", True
    End If
End Sub

Private Sub mnuFSetup_Click()
    Dim iLDW_hWnd As Long

    iLDW_hWnd = FindWindow(Class_LDW, vbNullString)
    If iLDW_hWnd = 0 Then Call OpenLDWobject

       ShowWindow iLDW_hWnd, SW_Minimum
       ShowWindow iLDW_hWnd, SW_Normal

       SendKeys "%FR", True
End Sub

Private Sub mnuFExit_Click()
    Call cmdExitBtn_Click
End Sub

Private Sub mnuVHier_Click()

    If Not mnuVImport.Enabled Then
       If VerifyLDWobjectConnection(APIfailedMSG) Then
          SendKeys "%W2", True
          SendKeys "%H{down}G", True
       End If
    Else
       Call DisplayMessage(NoFacilityComparedMSG, vbInformation, _
          vbOKOnly, conAPIcaption)
    End If
End Sub

Private Sub mnuVMatrix_Click()
```

```
   If VerifyLDWobjectConnection(APIfailedMSG) Then
      SendKeys "%VM~", True
      DoEvents
   End If
End Sub

Private Sub mnuVSensitivity_Click()

   If VerifyLDWobjectConnection(APIfailedMSG) Then
      SendKeys "%SY", True
   End If
End Sub

Private Sub mnuVStackBar_Click()

   If VerifyLDWobjectConnection(APIfailedMSG) Then
      SendKeys "%SB", True
      DoEvents
      SendKeys "~", True
   End If

End Sub

Private Sub mnuVImport_Click()

   If bSQLQueryDone Then
      tbararies.Buttons("import").Enabled = False
      mnuVImport.Enabled = False
      fraLDWcntrl.Visible = False

      bLDWactive = OpenLDWobject
      Call ImportLDWdata
      Call ViewLDWmatrix
   Else
      Call DisplayMessage(NoFacilityComparedMSG, vbInformation, _
         vbOKOnly, conAPIcaption)
   End If
End Sub

Private Sub mnuRPrintAll_Click()
   Dim iMsgBxResponse As Byte

   iMsgBxResponse = DisplayMessage(PrinterReadyMSG, vbExclamation, _
      vbOK, conAPIcaption)

   If iMsgBxResponse = vbOK Then
      If VerifyLDWobjectConnection(LDWactiveMSG) Then
         Call TimeDelay(Time$(), 2)
         Call ActivateLDWReports
         AppActivate ARIES_Application
      End If
   End If
End Sub

Private Sub txtPropFacID_Change()

   With txtPropFacID
      If (.Text = "") Or (Len(.Text) < 5) Then
         .Text = .List(0)
      End If
   End With

End Sub

Private Sub cboUIC_Change()

   With cboUIC
      If (.Text = "") Or (Len(.Text) < 6) Then
         .Text = .List(0)
```

```
      End If
   End With

End Sub

Private Sub cmdAcceptBtn_Click()
   Dim iFacIDctr As Byte
   Dim Index As Byte
   Dim i As Byte
   Dim iConfirmResponse As Byte
   Dim cMsg As String

   If bValidUIC Then
      iFacIDctr = CountFacIDs
      For i = 0 To 3
         If txtFacID(i).Tag = "NOT USED" Then
            Index = i
            Exit For
         End If
      Next i

      If iFacIDctr < 4 Then
         txtFacID(Index).Text = txtPropFacID.Text
      Else
         cMsg = "Maximum number of Facilities Selected." & vbCr & vbCr & _
            "Do You Wish to Delete Facility {" & txtFacID(3).Text & _
            "}?"
         iConfirmResponse = MsgBox(cMsg, vbQuestion + vbYesNo,
   conFacilityErrorCaption)
         If iConfirmResponse = vbYes Then
            txtFacID(3).Text = txtPropFacID.Text
         End If
      End If

   Else
      If Not cboUIC.Text = "-NONE-" Then
         txtMovUIC.Text = cboUIC.Text
      Else
         cMsg = "Facility {" & txtPropFacID & "} has no Units Assigned."
         Call DisplayMessage(cMsg, vbExclamation, vbOKOnly, conUICcaption)
      End If
   End If
End Sub

Private Sub cmdAriesBtn_Click()

   Call CompareFacilities

End Sub

Private Sub cmdClearBtn_Click()

   'Reset all controls and elements of the User Interface to Defaults
   Call InitFacilityElements              'Reset controls to Defaults
   Call InitMenuBar
   Call InitToolBar
   Call InitAriesControls
   Call InitStatusBar
   Call InitGIStreeview
   Call TrafficLight(vbRed)

   If AriesStatusBar.Panels(5).Text = "Custom Tool : FacID Info" Then
      cmdAcceptBtn.Enabled = True
   End If

   bMapBasicRunning = False
   bSQLQueryDone = False

   txtMovUIC.SetFocus                     'Set cursor at MOVUIC txtbox
   Refresh
```

```
End Sub

Public Sub cmdExitBtn_Click()
   cProcName = "ExitBtn_Click"

   Dim iConfirmResponse As Byte        'MSGBox Response Variable

   If bSQLQueryDone Then
      'Display Confirm CompactDatabase Messagebox
      iConfirmResponse = MsgBox(CompactDatabaseMSG, _
         vbQuestion + vbYesNo + vbApplicationModal, conFileCaption)

      'Evaluate User Response
      If iConfirmResponse = vbYes Then 'Compact Database
         Call CompactAriesDataBase
      End If
   End If

   If Not (objGeoSelect Is Nothing) Then
      objGeoSelect.SetCallback Nothing
   End If

   Set objGeoSelect = Nothing
   Set objCallback = Nothing
   Set objGeoQuery = Nothing
   Set objExcel = Nothing

   Set dbAcropolis = Nothing
   Set dbTeliko = Nothing

   Unload Me                           'Close User Interface

End Sub

Private Sub cmdYesBtn_Click()

   Call mnuVImport_Click
End Sub

Private Sub cmdNoBtn_Click()
   Dim iConfirmResponse As Byte
   Dim cMsg As String

   cMsg = "Do You Wish to Discard Facility Comparisons?"
   iConfirmResponse = MsgBox(cMsg, vbQuestion + vbYesNo, conSQLcaption)
   If iConfirmResponse = vbYes Then
      fraLDWcntrl.Visible = False
      Call InitMeasuresGrid(0)
      Call cmdClearBtn_Click
   End If
End Sub

Private Sub spinFacIDbtn_SpinDown()
   Dim iComboBoxMax As Integer

   With txtPropFacID
      iComboBoxMax = .ListCount - 1
      If Not (iPropFacIDctr = iComboBoxMax) Then
         iPropFacIDctr = iPropFacIDctr + 1
         .Text = .List(iPropFacIDctr)
         Call LoadPropFacilityUICBox(.Text)
         Call DisplayPropFacilityUnitNames
      End If
   End With
End Sub

Private Sub spinFacIDbtn_SpinUp()

   With txtPropFacID
      If Not iPropFacIDctr = 0 Then
```

```
            iPropFacIDctr = iPropFacIDctr - 1
            .Text = .List(iPropFacIDctr)
            Call LoadPropFacilityUICBox(.Text)
            Call DisplayPropFacilityUnitNames
         End If
      End With
End Sub

Private Sub tbarAries_ButtonClick(ByVal Button As Button)
   Dim iLDWhWnd As Long          'API window handler
   Dim iMsgBxResponse As Byte    'Messagebox response variable

   Select Case Button.Key
      Case "selector"
         objGeoSelect.RunMenuCommand M_TOOLS_SELECTOR
         AriesStatusBar.Panels(5).Text = "MapInfo Tool : Selector"

      Case "grabber"
         objGeoSelect.RunMenuCommand M_TOOLS_RECENTER
         AriesStatusBar.Panels(5).Text = "MapInfo Tool : Grabber"

      Case "expand"
         objGeoSelect.RunMenuCommand M_TOOLS_EXPAND
         AriesStatusBar.Panels(5).Text = "MapInfo Tool : Zoom-In"

      Case "shrink"
         objGeoSelect.RunMenuCommand M_TOOLS_SHRINK
         AriesStatusBar.Panels(5).Text = "MapInfo Tool : Zoom-Out"

      Case "infotool"
         objGeoSelect.Do "Run Menu Command ID 501"
         AriesStatusBar.Panels(5).Text = "Custom Tool : FacID Info"
         pnlFacilityInfo.Enabled = True
         cboUIC.Visible = True
         cboUIC.Clear
         txtPropFacID.Visible = True
         txtPropFacID.Clear
         cmdAcceptBtn.Enabled = True

      Case "mapinfo"
         If Not bMapInfoRunning Then
            iMapInfoWinID = Shell(FILE_MapInfo, vbNormalFocus)
            SendKeys "~", True
            SendKeys "%FL", True
            iMapInfoWinID = FindWindow(Class_MapInfo, vbNullString)
            If iMapInfoWinID = 0 Then
               Call DisplayMessage(APIfailedMSG, vbCritical, vbOKOnly, _
                  conAPIcaption)
            End If
            bMapInfoRunning = True
            mnuFCloseMap.Enabled = True
         Else
            ShowWindow iMapInfoWinID, SW_Minimum
            ShowWindow iMapInfoWinID, SW_Normal
            SendKeys "%FX", True
            bMapInfoRunning = False
            mnuFCloseMap.Enabled = False
         End If

      Case "ldw"
         iLDWhWnd = FindWindow(Class_LDW, vbNullString)
         If iLDWhWnd = 0 Then
            bLDWactive = OpenLDWobject
         Else
            iMsgBxResponse = DisplayMessage(LDWresetMSG, vbQuestion, vbOK, _
               conAPIcaption)
            If iMsgBxResponse = vbOK Then
               Call ResetLDWobject
               AppActivate ARIES_Application
               tbararies.Buttons("import").Enabled = True
```

```
                fraLDWcntrl.Visible = True
                mnuVImport.Enabled = True
            End If
         End If

      Case "import"
         Call mnuVImport_Click

      Case "file"

      Case "report"
         Call mnuRPrintAll_Click

      Case "matrix"
         Call mnuVMatrix_Click

      Case "hierarchy"
         Call mnuVHier_Click

      Case "dynamic"
         Call mnuVSensitivity_Click

      Case "stacked"
         Call mnuVStackBar_Click

      Case "about"

    End Select
End Sub

Private Sub cmdClearBtn_GotFocus()

   'Display CLEAR button hint on main status bar
   AriesStatusBar.Panels(1).Text = ClearButtonMSG
End Sub

Private Sub cmdAriesBtn_GotFocus()

   'Display GENERATE button hint on main status bar
   AriesStatusBar.Panels(1).Text = LaunchAriesMSG
End Sub

Private Sub cmdExitBtn_GotFocus()

   'Display EXIT Button hint on main status bar
   AriesStatusBar.Panels(1).Text = ExitButtonMSG
End Sub

Private Sub txtMovUIC_GotFocus()
   AriesStatusBar.Panels(1).Text = MovUICstatbarMSG
End Sub

Private Sub txtFacID_GotFocus(Index As Integer)
   AriesStatusBar.Panels(1).Text = FacilityStatbarMSG & CStr(Index + 1)
   lstbxFacID(Index).Visible = True
End Sub

Private Sub txtMovUIC_LostFocus()

   If (Len(txtMovUIC) = 0) And txtMovUIC.Tag = "VALID" Then
      Call TrafficLight(vbRed)
      bValidUIC = False
      txtMovUIC.Tag = "NOT USED"
      lstbxMovUIC.Clear
   End If
End Sub

Private Sub txtFacID_LostFocus(Index As Integer)
   Dim iFacIDctr As Byte
```

```
    If (Len(txtFacID(Index)) = 0) And txtFacID(Index).Tag = "VALID" Then
       txtFacID(Index).Tag = "NOT USED"
       lstbxFacID(Index).Clear
       lstbxFacID(Index).AddItem "* * DELETED * *"
    End If

    iFacIDctr = CountFacIDs
    If iFacIDctr = 0 And txtMovUIC.Tag = "VALID" Then
       Call TrafficLight(vbYellow)
    ElseIf txtMovUIC.Tag = "NOT USED" Then
       Call TrafficLight(vbRed)
    End If
End Sub

Private Sub txtFacID_Change(Index As Integer)
    Dim bValidFacID As Boolean
    Dim bValidUnit As Boolean
    Dim cFacID As String * 5
    Dim cMsgText As String
    Dim rsValidateFacID As Recordset
    Dim rsValidateUnit As Recordset

    cMsgText = ""
    bValidFacID = False
    bValidUnit = False

    'Open Validation Table
    Set rsValidateFacID = dbAcropolis.OpenRecordset(Tbl_ValidateUIC, dbOpenTable,
    dbReadOnly)
    rsValidateFacID.Index = "FACID"

    If bGEOREF_Exists Then
       Set rsValidateUnit = dbAcropolis.OpenRecordset(Tbl_ValidateUnit,
    dbOpenTable, dbReadOnly)
       rsValidateUnit.Index = "FACID"
    End If

    'Automatically determine if FACID entered
    If Len(txtFacID(Index)) = 5 Then
       cFacID = txtFacID(Index).Text
       bValidFacID = ValidateFacID(cFacID, rsValidateFacID)
       If bGEOREF_Exists Then
          bValidUnit = ValidateFacID(cFacID, rsValidateUnit)
       End If

       If bValidFacID Or bValidUnit Then     'Valid FACID exists
          If bValidFacID Then
             Call LoadFacilityListBox(Index, rsValidateFacID, 0)
          Else
             Call LoadFacilityListBox(Index, rsValidateUnit, 1)
          End If

          aFacID(Index + 1) = cFacID
          txtFacID(Index).Tag = "VALID"

          If Not (Index = 3) Then      'Prep next Facility Frame
             Call InitializeFacilityFrame(Index + 1)
             Call TrafficLight(vbGreen)
          Else                         'Last Facility Frame used
             cmdAriesBtn.SetFocus
          End If

       Else                            'Facility ID invalid
          cMsgText = "FacID: {" & cFacID & FacIDinvalidMSG
          Call DisplayMessage(cMsgText, vbInformation, vbOKOnly,
    conFacilityErrorCaption)
          txtFacID(Index).Text = ""
          txtFacID(Index).Tag = "NOT USED"
       End If
```

```
    Else
        AriesStatusBar.Panels(1).Text = FacIDlengthMSG
    End If

    'Delete Recordset Object
    rsValidateFacID.Close
    Set rsValidateFacID = Nothing
    rsValidateUnit.Close
    Set rsValidateUnit = Nothing

End Sub

Private Sub txtMovUIC_Change()
    cProcName = "TXTMOVUIC_CHANGE"
    Dim cMsgText As String

    cMsgText = ""
    bValidUIC = False

    If Len(txtMovUIC) = 6 Then
        cMovUic.UIC = txtMovUIC.Text
        bValidUIC = ValidateMovUIC(cMovUic.UIC)

        If bValidUIC Then
            aFacID(0) = cMovUic.FacID
            Call LoadMovUIClistbox(cMovUic)
            Call InitializeFacilityFrame(0)
            cmdClearBtn.Enabled = True
            txtMovUIC.Tag = "VALID"

            'Set the appropiate data integrity light
            If Not txtFacID(0).Tag = "VALID" Then
                Call TrafficLight(vbYellow)
            Else
                Call TrafficLight(vbGreen)
            End If

        Else          'Unit ID Code invalid
            cMsgText = "UIC: {" & cMovUic.UIC & MovUICinvalidMSG
            Call DisplayMessage(cMsgText, vbInformation, vbOKOnly, conUICcaption)
            txtMovUIC.Text = ""
            txtMovUIC.Tag = "NOT USED"
        End If

    Else
        AriesStatusBar.Panels(1).Text = MovUIClengthMSG
    End If

End Sub

Private Sub txtFacID_KeyPress(Index As Integer, KeyAscii As Integer)
    KeyAscii = KeyUpperNumeric(KeyAscii)

End Sub

Private Sub txtMovUIC_KeyPress(KeyAscii As Integer)
    KeyAscii = KeyUpperNumeric(KeyAscii)

End Sub

Private Function CountFacIDs() As Byte
    Dim i As Byte                       'Loop Variable

    CountFacIDs = 0                     'Initialize function
    For i = 0 To 3                      'Loop thru Facility choices
        'Evaluate Facility Frame TAG field for Confirmed Selections
        If (txtFacID(i).Tag = "VALID") Then
            CountFacIDs = CountFacIDs + 1 'increment counter
        End If
    Next i
```

```
End Function

Private Sub Form_Load()

    'Establish DEFAULT position for User interface
    Me.Top = 0
    Me.Left = 0

    'Initialize Acropolis Database
    Set dbWorkspc = DBEngine.Workspaces(0)
    Set dbAcropolis = dbWorkspc.OpenDatabase(App.Path & FILE_Acropolis)
    'Set dbTeliko = dbWorkspc.OpenDatabase(App.Path & FILE_Teliko)
     Set dbMeasures = Workspaces(0).OpenDatabase(App.Path & DB_MEASURES)
     Set rsMeasures = dbMeasures.OpenRecordset("Measure Values", dbOpenTable)
     rsMeasures.Index = "FAC_ID"
     'Set rsProblems = dbMeasures.OpenRecordset("Problems", dbOpenTable)

    'Initialize the USER INTERFACE and Form Controls
    Call InitMenuBar
    Call InitToolBar
    Call InitMaps

    Call InitFacilityInfoPanel
    Call InitFacilityElements                    'Initialize Default values
    Call InitGIStreeview

    Call TrafficLight(vbRed)
    Call InitAriesControls
    Call InitStatusBar

    Me.Show                                      'Display form to User
    Me.WindowState = vbMaximized
    Refresh

    Screen.MousePointer = vbDefault              'reset mousepointer to arrow
    txtMovUIC.SetFocus                           'set cursor at MOVUIC txtbox

End Sub

Private Sub InitializeFacilityFrame(ByVal Index As Integer)
    cProcName = "INITIALIZE_FACILITY_FRAME"

    If Not Index = 4 Then
        fraFacility(Index).Enabled = True
        With txtFacID(Index)
            .Visible = True
            .Enabled = True
            .SetFocus
        End With
    End If
End Sub

Private Sub InitMenuBar()
    cProcName = "INIT_MENU_BAR"

    mnuVImport.Enabled = True
    mnuFCloseLDW.Enabled = True

End Sub

Private Sub InitToolBar()
    cProcName = "INIT_TOOL_BAR"

        tbararies.Buttons("import").Enabled = True
        tbararies.Buttons("report").Enabled = True
        tbararies.Buttons("matrix").Enabled = True
        tbararies.Buttons("hierarchy").Enabled = True
        tbararies.Buttons("dynamic").Enabled = True
        tbararies.Buttons("stacked").Enabled = True
End Sub
```

```
Private Sub InitMaps()
   cProcName = "INIT_MAPS"
   Dim bMapSelectActive As Boolean

   'Load MapInfo Objects for GeoQuery Display
   bMapQueryActive = ConnectOLEobject(objGeoQuery, "Mapinfo.Application")
   If bMapQueryActive Then      'Ensure Application loaded
      objGeoQuery.Do "Set Application Window " & pictMapQueryFrame.hwnd
      objGeoQuery.Do "Set Next Document Parent " & pictMapQueryFrame.hwnd & "
    Style 1 "
   End If

   'Load MapInfo Objects for GeoSelect Display
   bMapSelectActive = ConnectOLEobject(objGeoSelect, "Mapinfo.Application")
   If bMapSelectActive Then      'Ensure Application loaded
      objGeoSelect.Do "Set Application Window " & pictMapSelectFrame.hwnd
      objGeoSelect.Do "Set Next Document Parent " & pictMapSelectFrame.hwnd & "
    Style 1 "
   End If

   'Create an Instance of the OLE Callback Object
   Set objCallback = New clsOLECallback
   objGeoSelect.SetCallback objCallback

   'Load & Display the Maps with labels
   Call OpenUS_StatesLayer
   Call OpenUSARC_GeoRefLayer
   Call OpenStateCapitalsLayer
   Call OpenMajorCityLayer
   Call DisplayMapLabels
   Call PositionMap

   'Modify the MapInfo Right-Click Menu
   Call InitializeMapInfoMenu
   Call InitializeMapInfoToolbar
   Call InitializeMapInfoStatusBar

   'Set Zoom-In tool as the Default
   objGeoSelect.RunMenuCommand M_TOOLS_EXPAND
   AriesStatusBar.Panels(5).Text = "MapInfo Tool : Zoom-In"
   Refresh
End Sub

Private Sub InitMeasuresGrid(ByVal Index As Byte)
   cProcName = "INIT_MEASURES_GRID"
   Dim i As Integer
   Dim iGridColumnWidth(conMaximumColumns) As Integer
   Dim cGridRowLabel(conTotalMeasures) As String

   'Load Column Width array
   iGridColumnWidth(0) = conOneColumnWidth
   iGridColumnWidth(1) = conTwoColumnWidth
   iGridColumnWidth(2) = conThreeColumnWidth
   iGridColumnWidth(3) = conFourColumnWidth
   iGridColumnWidth(4) = conFiveColumnWidth

   'Load Row Labels array
   cGridRowLabel(0) = conMeasure0
   cGridRowLabel(1) = conMeasure1
   cGridRowLabel(2) = conMeasure2
   cGridRowLabel(3) = conMeasure3
   cGridRowLabel(4) = conMeasure4
   cGridRowLabel(5) = conMeasure5
   cGridRowLabel(6) = conMeasure6
   cGridRowLabel(7) = conMeasure7
   cGridRowLabel(8) = conMeasure8
   cGridRowLabel(9) = conMeasure9
   cGridRowLabel(10) = conMeasure10
   cGridRowLabel(11) = conMeasure11
```

```
    cGridRowLabel(12) = conMeasure12
    cGridRowLabel(13) = conMeasure13
    cGridRowLabel(14) = conMeasure14
    cGridRowLabel(15) = conMeasure15
    cGridRowLabel(16) = conMeasure16
    cGridRowLabel(17) = conMeasure17
    cGridRowLabel(18) = conMeasure18
    cGridRowLabel(19) = conMeasure19
    cGridRowLabel(20) = conMeasure20

    With gridMeasures
       'Establish Grid default parameters
       .Visible = True
       .Font.Size = 8

       'Reset Grid if data present
       .Cols = 1
       .Cols = 2
       .FixedCols = 1
       .FixedRows = 1

       'Set rows and columns.
       .Rows = 21
       .Cols = Index + 2

       'Set Fixed Column Label size
       .ColWidth(0) = 1370

       'Set remaining columns width dependent on number of Facilities
       For i = 1 To Index + 1
          .ColWidth(i) = iGridColumnWidth(Index)
          .ColAlignment(i) = 1
       Next i

       'Display Grid Row Labels in Column 1
       gridMeasures.Col = 0
       For i = 0 To 20
          gridMeasures.Row = i
          gridMeasures.Text = cGridRowLabel(i)
       Next i
    End With
End Sub

Private Sub InitFacilityInfoPanel()
    cProcName = "INIT_FACILITY_INFO_PANEL"

    iPropFacIDctr = 0

    pnlFacilityInfo.Enabled = False
    txtPropFacID.Visible = False
    cboUIC.Visible = False
    spinFacIDbtn.Visible = False
    Refresh
End Sub

Private Sub InitFacilityElements()
    cProcName = "INIT_FACILITY_ELEMENTS"

    Dim i As Byte                          'Loop Variable

    cMovUic.UIC = ""                       'Set MOVUIC variable to NULL
    cMovUic.FacID = ""

    tabAriesMap.Tab = 0                    'Display GeoSelection Map
    tabAriesUI.Tab = 0                     'Display Facility Select Tab

    fraMovUIC.Tag = "NOT USED"             'Initialize Facility Frame TAGs
    fraMovUIC.Font.Size = 10               'Set Font Size to default
    txtMovUIC.Text = ""                    'Clear MOVUIC text box
    txtMovUIC.Enabled = True               'Allow user input
```

```
    lstbxMovUIC.Clear                       'Empty listbox

    For i = 0 To 3                          'Loop thru Facility Frames
       fraFacility(i).Enabled = False       'Disable Frame
       fraFacility(i).Font.Size = 9         'Set Font Size to default
       txtFacID(i).Text = ""                'Clear facility text box
       txtFacID(i).Visible = False          'Hide facility text box
       lstbxFacID(i).Clear                  'Empty facility list box
       lstbxFacID(i).Visible = False        'Hide facility list box
       txtFacID(i).Tag = "NOT USED"         'Set Frame TAG to default
    Next i

    For i = 0 To 4                          'Loop thru SQL status frames
       fraStatus(i).Enabled = False         'Disable status frame
       fraStatus(i).Font.Size = 10          'Set frame font default
       txtStatus(i).Text = ""               'Clear status text box
       txtStatus(i).Visible = True          'Hide status text box
       txtStatus(i).Enabled = True
       lstbxStatus(i).Clear                 'Empty status listbox
       lstbxStatus(i).Visible = False       'Hide status list box
       aFacID(i) = ""
    Next i

    txtPropFacID.Clear
    cboUIC.Clear
    lstbxUnitName.Clear
    txtCity.Text = ""
    txtState.Text = ""
    txtZipCode.Text = ""

    fraStatus(MeasuresFrameIndex).Enabled = True
    barSQLstatus.Value = 0                  'Set progress bar to Zero
    iProgressIndicator = 0                  'Set progress indicator default
    barSQLstatus.Min = 0                    'Set Progress bar minimum
    Refresh
End Sub

Private Sub InitGIStreeview()
    cProcName = "INIT_GIS_TREEVIEW"

    trvGISData.Nodes.Clear
    trvGISData.Enabled = False
    Refresh
End Sub

Private Sub InitAriesControls()
    cProcName = "INIT_ARIES_CONTROLS"

    cmdAriesBtn.Enabled = False             'Disable Measures Button
    cmdClearBtn.Enabled = False             'Disable CLEAR button
    cmdAcceptBtn.Visible = True
    cmdAcceptBtn.Enabled = False            'Disable ACCEPT Button
    cmdAcceptBtn.Enabled = True
    cmdAcceptBtn.Font.Size = 12
    Refresh
End Sub

Private Sub InitStatusBar()
    cProcName = "INIT_STATUS_BAR"

    AriesStatusBar.Panels(2).Text = ""
    AriesStatusBar.Panels(3).Text = ""
    Refresh
End Sub

Private Sub FormElementsLock(ByVal iFacIndex As Byte)
    cProcName = "FORM_ELEMENTS_LOCK"
    Dim i As Byte                           'Loop Variable

    'Disable Query and Clear user entry buttons
```

```
    Call TrafficLight(vbBlack)
    cmdClearBtn.Enabled = False
    cmdAcceptBtn.Enabled = False

    txtMovUIC.Enabled = False
    lstbxMovUIC.Enabled = False

    'Locks User input until Query finished
    For i = 0 To 3
       txtFacID(i).Enabled = False
       lstbxFacID(i).Enabled = False
    Next i

    'Hides Exposed Facility Frame w/out User Input
    'If Not (iFacIndex = 4) Then
    '   txtFacID(iFacIndex).Visible = False
    '   lstbxFacID(iFacIndex).Visible = False
    '   fraFacility(iFacIndex).Enabled = False
    'End If

    'Bring Aries Measures Computation Tab to the Front
    tabAriesUI.Tab = 1
    Refresh
End Sub

Private Sub InitializeProgressBar(ByVal iFacIndex As Byte)
    cProcName = "PROGRESS_BAR_INITIALIZE"
    Dim iProgressIndicator As Integer

    'Initialize Progress Bar
    iStatusBarMax = ((iFacIndex + 1) * conTotalMeasures) + _
       ((iFacIndex + 1) * conTotalInterimTables) + conSingleMessages

    iProgressIndicator = 0
    With barSQLstatus
       .Min = 0
       .Value = 0
       .Max = iStatusBarMax
    End With
    Refresh
End Sub

Private Sub DisplayQueryStartTime(ByVal cStartTime As String)
    cProcName = "DISPLAY_QUERY_START_TIME"

    With AriesStatusBar
       .Panels(1).Text = LaunchMapinfoMSG
       cQueryStartTime = cStartTime
       .Panels(2).Text = "Start " & cQueryStartTime
    End With
    Refresh
End Sub

Private Sub DisplayQueryEndTime(ByVal cQueryEndTime As String)
    cProcName = "DISPLAY_QUERY_END_TIME"

    'Display Query End Time
    With AriesStatusBar
       .Panels(2).Text = "End " & cQueryEndTime
       Call ElapsedTimer(cQueryEndTime)
    End With
    Refresh
End Sub

Private Sub CompareFacilities()
    cProcName = "COMPARE_FACILITIES"

    Dim i As Byte                              'Loop Variable
    Dim j As Byte                              'Loop Variable
    Dim iFacIDctr As Byte
```

```
Dim iConfirmResponse As Integer
Dim cAriesConfirmMessage As String
Dim bDuplicateFacility As Boolean
Dim bSingleFacility As Boolean
Dim aMeasures(4) As Measures_Type
Dim rsFacilities As Recordset

Set rsFacilities = dbAcropolis.OpenTable("VALID_UNIT", dbOpenTable)

bDuplicateFacility = False
bSingleFacility = False
cAriesConfirmMessage = AriesConfirmMSG & vbCr

cMovUic.UIC = " "

'Function determines the Total number of Facilities Proposed
iFacIDctr = 100

'Determine if Proposed Facilities meet criterium (2 minimum)
'If iFacIDctr = 0 Then
'   bSingleFacility = True
'   Call InitFacilityElements
'   txtMovUIC.Text = cMovUic.UIC
'Else                                'Check if Duplicate Facility Selected
'   For i = 0 To (iFacIDctr - 1)
'      For j = (i + 1) To iFacIDctr
'         If aFacID(i) = aFacID(j) Then bDuplicateFacility = True
'      Next j
'   Next i
'End If

'Build Confirmation message
'For i = 0 To iFacIDctr
'   cAriesConfirmMessage = cAriesConfirmMessage & Space(4) & aFacID(i) & ","
'Next i
'cAriesConfirmMessage = Left$(cAriesConfirmMessage,
(Len(cAriesConfirmMessage) - 1))
cAriesConfirmMessage = "This will conduct a calculation of the Valid
Facilities in the Valid Unit List"
'If bDuplicateFacility Then          'Duplicate facility exists
'   Call DisplayMessage(DuplicateFacilityMSG, vbCritical, vbOKOnly,
conFacilityErrorCaption)

'ElseIf bSingleFacility Then         'Single facility exists
'   Call DisplayMessage(InvalidGISQueryMSG, vbCritical, vbOKOnly,
conFacilityErrorCaption)
'End If

If Not bDuplicateFacility And Not bSingleFacility Then
   Screen.MousePointer = vbHourglass
   iConfirmResponse = MsgBox(cAriesConfirmMessage, vbQuestion + vbOKCancel, _
      "Launch ARIES v3.0")

   If iConfirmResponse = vbOK Then
      Call FormElementsLock(iFacIDctr)
      Call DisplayQueryStartTime(Time$())

      Call InitializeProgressBar(iFacIDctr)
     'Call CreateMOS_InterestTable(cMovUic.UIC)

     With rsFacilities
         .MoveFirst

            'Loop thru all proposed facilities
         For i = 0 To iFacIDctr

         aMeasures(i).FacID = .Fields("FAC_ID").Value    'aFacID(i)
'         Call GetArchivedFacility(aMeasures(i), cMovUic.UIC)

'         If Not aMeasures(i).Archived Then
```

```
                Call NonGISQuery(aMeasures(i))
'              End If

              Call GISQuery(aMeasures(i))
              Call ResetSQLframes
              Call ArchiveFacilityMeasures(aMeasures(i))

              .MoveNext
            Next i
           End With
            Call CloseGISQuery
            Call DeleteTempAccessTables
            Call DisplayQueryEndTime(Time$())
'           Call OutputComputedMeasures(aMeasures(), iFacIDctr)

            tbararies.Buttons("import").Enabled = True
            cmdClearBtn.Enabled = True
            bSQLQueryDone = True
         End If
      End If
      Screen.MousePointer = vbDefault
End Sub

Private Sub GetArchivedFacility(ByRef aMeasures As Measures_Type, ByVal
    cMovUic As String)
    cProcName = "Get_Archived_Facility"
    Dim rsArchivedFacility As Recordset
    Dim QdefnSQL As String

    QdefnSQL = "Select * From Repository Where FAC_ID = '" & aMeasures.FacID & "'
    Order by MOVUIC"
    AriesStatusBar.Panels(1).Text = ArchivedFacilityMSG
    Refresh

    'Open Facility Respository Database
    Set rsArchivedFacility = dbTeliko.OpenRecordset(QdefnSQL, dbOpenSnapshot,
    dbReadOnly)

    With rsArchivedFacility
       If TableNotNULL(rsArchivedFacility) Then
          .MoveFirst
          AriesStatusBar.Panels(1).Text = LoadingArchiveMSG & !Fac_ID
          Refresh
          'Transfer Facility Record to Measures table
          aMeasures.Archived = True
          Call DisplayStatus(conMeasure1, NonGISFrameIndex)
          aMeasures.FacBacklogdMaint = !FAC_MAINT       'Measure #1
          Call DisplayStatus(conMeasure2, NonGISFrameIndex)
          aMeasures.OperatingCost = !FAC_OPCOST         'Measure #2
          Call DisplayStatus(conMeasure3, NonGISFrameIndex)
          aMeasures.FacAge = !FAC_AGE                   'Measure #3
          Call DisplayStatus(conMeasure4, NonGISFrameIndex)
          aMeasures.FacCond = !FAC_COND                 'Measure #4
          Call DisplayStatus(conMeasure5, NonGISFrameIndex)
          aMeasures.FacOwned = !FAC_OWNED               'Measure #5
          Call DisplayStatus(conMeasure6, MeasuresFrameIndex)
          aMeasures.Competition = !COMPETION            'Measure #6
          Call DisplayStatus(conMeasure7, MeasuresFrameIndex)
          aMeasures.AreaDrillAttend = !DRILL_ATND       'Measure #7
          Call DisplayStatus(conMeasure8, MeasuresFrameIndex)
          aMeasures.AreaLossRate = !AREA_LOSS           'Measure #8
          Call DisplayStatus(conMeasure9, MeasuresFrameIndex)
          aMeasures.AreaXferRate = !AREA_XFER           'Measure #9
          Call DisplayStatus(conMeasure10, MeasuresFrameIndex)
          aMeasures.AvgAreaMan = !AVG_MANING            'Measure #10
          Call DisplayStatus("RZA Distance", GISNonSQLFrameIndex)
          aMeasures.DistToRecruit = !DIST_RZA           'Measure #11
          Call DisplayStatus(conMeasure12, MeasuresFrameIndex)
          aMeasures.TotalAvailClos = !AVAIL_CLOS        'Measure #12
          Call DisplayStatus(conMeasure13, MeasuresFrameIndex)
```

```
            aMeasures.IRR = !IRR                          'Measure #13
            Call DisplayStatus(conMeasure14, MeasuresFrameIndex)
            aMeasures.RecruitMarket = !REC_MARKET         'Measure #14
            Call DisplayStatus(conMeasure15, MeasuresFrameIndex)
            aMeasures.Reassignments = !REASSIGN           'Measure #15
            Call DisplayStatus("AMSA Distance", GISNonSQLFrameIndex)
            aMeasures.DistToAMSA = !DIST_AMSA             'Measure #16
            Call DisplayStatus("ECS Distance", GISNonSQLFrameIndex)
            aMeasures.DistToECS = !DIST_ECS               'Measure #17
            Call DisplayStatus(conNoAuthNG, GISNonSQLFrameIndex)
            Call DisplayStatus(vbNullString, GISNonSQLFrameIndex)
            Call DisplayStatus(conMeasure18, NonGISFrameIndex)
            Call DisplayStatus(vbNullString, NonGISFrameIndex)
            aMeasures.FacWkndUsed = !WKND_USED            'Measure #18
            Call IncrementProgressBar
            Call IncrementProgressBar

            .FindFirst "MOVUIC = '" & cMovUic & "'"
            If Not .NoMatch Then
               aMeasures.MovUIC_Match = True
               Call DisplayStatus(conMeasure19, MeasuresFrameIndex)
               aMeasures.MOSAvailClos = !MOS_AVAIL        'Measure #19
               Call DisplayStatus(conMeasure20, MeasuresFrameIndex)
               aMeasures.IRR_MOS = !IRR_MOS               'Measure #20
               'Increment Progress Bar to account for No Temp Tables created
               iStatusBarMax = iStatusBarMax - 12
               barSQLstatus.Max = iStatusBarMax
            Else
               aMeasures.MovUIC_Match = False
            End If
         End If
         .Close
      End With
      Set rsArchivedFacility = Nothing
   End Sub

   Private Sub CreateMOS_InterestTable(ByVal cMovUic As String)
      cProcName = "CREATE_MOS_INTEREST_TABLE"
      Dim cQdefnSQL As String
      Dim bNoASSNxMOS_Exists As Boolean

      AriesStatusBar.Panels(1).Text = BuildMOSInterestMSG

      bNoASSNxMOS_Exists = CreateNoASSNxMOStable(cMovUic)
      If bNoASSNxMOS_Exists Then
         'Create Interim Table [ MOS_INTEREST ]
         Call DisplayStatus(Tbl_MOSInterest, SQLFrameIndex)
         cQdefnSQL = GetMOSInterestQuery
         Call BuildAccessTable(Tbl_MOSInterest, Qdef_MOSInterest, cQdefnSQL)
      End If
   End Sub

   Private Function CreateNoASSNxMOStable(ByVal cMovUic As String) As Boolean
      Dim cQdefnSQL As String

      'Create Interim Table [ NoASSNxMOS ]
      cQdefnSQL = NoASSNxMOSQueryBegin & cMovUic & NoASSNxMOSQueryEnd
      Call DisplayStatus(Tbl_NoASSNxMOS, SQLFrameIndex)
      Call BuildAccessTable(Tbl_NoASSNxMOS, Qdef_NoASSNxMOS, cQdefnSQL)
      CreateNoASSNxMOStable = True

   End Function

   Private Function GetMOSInterestQuery() As String
      cProcName = "GET_MOS_INTEREST_QUERY"

      Dim rsMOS_Total As Recordset
      Dim rsMOS_Top3 As Recordset

      Dim iRecordCtr As Long
```

```
    Dim iMOS_Total As Integer
    Dim iMOS_Top3 As Integer
    Dim MOSPercent As Single

    MOSPercent = 0.5
    iMOS_Total = 0
    iMOS_Top3 = 0

On Error GoTo EH_MOSinterest

    'Execute SQL Query for MOS_Count Summation
    Set rsMOS_Total = dbAcropolis.OpenRecordset(MOS_TotalQuery, dbOpenDynaset,
    dbReadOnly)
    If QuerySumNotNULL(rsMOS_Total!MOS_TOTAL) Then
       iMOS_Total = rsMOS_Total!MOS_TOTAL
    Else
       iMOS_Total = -1
    End If

    'Execute SQL Query for MOS_Top 3
    Set rsMOS_Top3 = dbAcropolis.OpenRecordset(MOS_Top3Query, dbOpenDynaset,
    dbReadOnly)

    'Sum Top 3 MOS's
    With rsMOS_Top3
       If TableNotNULL(rsMOS_Top3) Then .MoveLast
       iRecordCtr = .RecordCount

       If TableNotNULL(rsMOS_Top3) And iRecordCtr = 3 Then
          .MoveFirst
          iMOS_Top3 = !MOS_COUNT

          .MoveNext
          iMOS_Top3 = iMOS_Top3 + !MOS_COUNT

          .MoveLast
          iMOS_Top3 = iMOS_Top3 + !MOS_COUNT
       Else
          iMOS_Top3 = 1
       End If
    End With

    'Determine MOS of Interest
    If (CSng(iMOS_Top3) / CSng(iMOS_Total)) > MOSPercent Then
       GetMOSInterestQuery = MOS_INT_Top3Query
    Else
       GetMOSInterestQuery = MOS_INT_TotalQuery
    End If

    'Delete Recordset Objects
    rsMOS_Total.Close
    rsMOS_Top3.Close
    Set rsMOS_Total = Nothing
    Set rsMOS_Top3 = Nothing

Exit Function

EH_MOSinterest:

    Select Case Err.Number

       Case ERR_InvalidUseOFNull
          iMOS_Total = -1
          Resume Next

       Case Else   'Trap all other Errors & Report
          Call OutputERROR_LOG
          End

    End Select
```

```
End Function

Private Sub ResetSQLframes()

   lstbxStatus(MeasuresFrameIndex).Clear
   txtStatus(MeasuresFrameIndex).Text = ""

   lstbxStatus(NonGISFrameIndex).Clear
   txtStatus(NonGISFrameIndex).Text = ""

   lstbxStatus(GISNonSQLFrameIndex).Clear
   txtStatus(GISNonSQLFrameIndex).Text = ""

   lstbxStatus(SQLFrameIndex).Clear
   txtStatus(SQLFrameIndex).Text = ""
End Sub

Private Sub NonGISQuery(ByRef aMeasures As Measures_Type)
   cProcName = "NON_GIS_QUERY"

   Dim rsFacBklogdMaintMeasure As Recordset  'Measure #1
   Dim rsOpCostFacCondMeasure As Recordset   'Measure #2 & #4
   Dim rsFacAgeMeasure As Recordset          'Measure #3
   Dim rsFacOwnedUsedMeasure As Recordset    'Measure #5 & #18

   Set rsFacBklogdMaintMeasure = dbAcropolis.OpenRecordset(Tbl_RPINFODT,
   dbOpenTable, dbReadOnly)
   rsFacBklogdMaintMeasure.Index = "FACID"

   Set rsOpCostFacCondMeasure = dbAcropolis.OpenRecordset(Tbl_FPS, dbOpenTable,
   dbReadOnly)
   rsOpCostFacCondMeasure.Index = "FACID"

   Set rsFacAgeMeasure = dbAcropolis.OpenRecordset(Tbl_Interest, dbOpenTable,
   dbReadOnly)
   rsFacAgeMeasure.Index = "FACID"

   Set rsFacOwnedUsedMeasure = dbAcropolis.OpenRecordset(Tbl_Complex,
   dbOpenTable, dbReadOnly)
   rsFacOwnedUsedMeasure.Index = "FACID"

   'Calc Measure #1
   With rsFacBklogdMaintMeasure
      .Seek "=", aMeasures.FacID
      If Not (.NoMatch) Then
         aMeasures.FacBacklogdMaint = !Maint_Cost
      Else
         aMeasures.FacBacklogdMaint = DefaultErrorValue
      End If
      Call DisplayStatus(conMeasure1, NonGISFrameIndex)
   End With

   'Calc Measure #2 & #4
   rsOpCostFacCondMeasure.Seek "=", aMeasures.FacID
   If Not (rsOpCostFacCondMeasure.NoMatch) Then
      aMeasures.OperatingCost = rsOpCostFacCondMeasure!COST_PR_SF

      'Verify that the stored value is not NULL
      If Not IsNull(rsOpCostFacCondMeasure!FAC_COND) Then
         aMeasures.FacCond = rsOpCostFacCondMeasure!FAC_COND
      Else
         aMeasures.FacCond = DefaultErrorValue
      End If
   Else
      aMeasures.OperatingCost = DefaultErrorValue
      aMeasures.FacCond = DefaultErrorValue
   End If
   Call DisplayStatus(conMeasure2, NonGISFrameIndex)
   Call DisplayStatus(conMeasure4, NonGISFrameIndex)
```

```
    'Calc Measure #3
    With rsFacAgeMeasure
       .Seek "=", aMeasures.FacID
       If Not (.NoMatch) Then
          If Not IsNull(!DATE_ACQ) Then
             aMeasures.FacAge = ComputeFacilityAge(!DATE_ACQ)
          Else
             aMeasures.FacAge = DefaultErrorValue
          End If
       Else
         aMeasures.FacAge = DefaultErrorValue
       End If
       Call DisplayStatus(conMeasure3, NonGISFrameIndex)
    End With

    'Calc Measure #5 & # 18
    rsFacOwnedUsedMeasure.Seek "=", aMeasures.FacID
    If Not (rsFacOwnedUsedMeasure.NoMatch) Then
       If Not IsNull(rsFacOwnedUsedMeasure!FAC_OWNED) Then
          aMeasures.FacOwned = rsFacOwnedUsedMeasure!FAC_OWNED
       Else
          aMeasures.FacOwned = DefaultErrorValue
       End If
       aMeasures.FacWkndUsed = rsFacOwnedUsedMeasure!FAC_WKND_USED
    Else
       aMeasures.FacOwned = DefaultErrorValue
       aMeasures.FacWkndUsed = DefaultErrorValue
    End If
    Call DisplayStatus(conMeasure5, NonGISFrameIndex)
    Call DisplayStatus(conMeasure18, NonGISFrameIndex)
    Call DisplayStatus(vbNullString, NonGISFrameIndex)

'Close Recordset Objects
    Set rsFacBklogdMaintMeasure = Nothing
    Set rsOpCostFacCondMeasure = Nothing
    Set rsFacAgeMeasure = Nothing
    Set rsFacOwnedUsedMeasure = Nothing
End Sub

Private Sub GISQuery(ByRef aMeasures As Measures_Type)
    cProcName = "GIS_QUERY"
    Dim cQdefnSQL As String
    Dim rsArchive As Recordset
    Dim rsNoREQD As Recordset
    Dim rsNoAuthNG As Recordset
    Dim rsNoASSN As Recordset
    Dim rsNoLOSS As Recordset
    Dim rsNoXFER As Recordset
    Dim rsDrillTotal As Recordset
    Dim rsDrillSAT As Recordset
    Dim rsAvailClos As Recordset
    Dim rsIRR As Recordset
    Dim rsRecruitMarket As Recordset
    Dim rsReassignments As Recordset
    Dim rsMOSAvailClos As Recordset
    Dim rsIRR_MOS As Recordset

    'Display Statusbar message
    AriesStatusBar.Panels(1).Text = SQLStatusMSG & aMeasures.FacID

'    If Not aMeasures.Archived Then
       'Execute MapInfo GeoQuery via OLE interface
       Call DisplayStatus(aMeasures.FacID, MapinfoFrameIndex)
       If bMapQueryActive Then
          Call MapInfoQuery(aMeasures.FacID)
       Else
          Call DisplayMessage(OLEfailedMSG, vbCritical, vbOKOnly, _
             conOLEcaption)
          bMapBasicRunning = False
```

```
      Call cmdExitBtn_Click
   End If
   Call ElapsedTimer(Time$())

   'Get calculated AMSA, ECS, & RZA Distances from proposed Facility
   Call GetFacilityDistances(aMeasures)          'Measure #11, 16, 17

   'Build Temp Tables FinanceCY, AreaFacID, AreaZipCode, AreaUIC, AreaClosUIC
   Call CreateFinanceTempTable
   Call CreateAreaFacIDTempTable
   Call CreateAreaUICTempTable
   Call CreateAreaClosUICTempTable
   Call CreateAreaZipCodeTempTable
   Call CreateAreaGL8ZipTempTable
   Call CreateAreaGL8M0STempTable
   Call DisplayStatus(vbNullString, SQLFrameIndex)

   'Calculate Measure #6
   Call DisplayStatus(conMeasure6, MeasuresFrameIndex)
   Set rsNoAuthNG = dbAcropolis.OpenRecordset(ExTbl_AreaDistance,
dbOpenSnapshot, dbReadOnly)
   rsNoAuthNG.MoveFirst
   Call DisplayStatus(conNoAuthNG, GISNonSQLFrameIndex)
   Call DisplayStatus(vbNullString, GISNonSQLFrameIndex)

   Set rsNoREQD = dbAcropolis.OpenRecordset(NoReqdQuery, dbOpenSnapshot,
dbReadOnly)
   If QuerySumNotNULL(rsNoREQD!TOTAL_REQD) Then
      aMeasures.Competition = rsNoAuthNG!NO_AUTH_NG + rsNoREQD!TOTAL_REQD
   Else
      aMeasures.Competition = DefaultErrorValue
   End If

   'Calculate Measure #7
   Call DisplayStatus(conMeasure7, MeasuresFrameIndex)

   Set rsDrillTotal = dbAcropolis.OpenRecordset(DrillTotalQuery,
dbOpenSnapshot, dbReadOnly)
   Call ElapsedTimer(Time$())

   Set rsDrillSAT = dbAcropolis.OpenRecordset(DrillSATQuery, dbOpenSnapshot,
dbReadOnly)

   If QuerySumNotNULL(rsDrillTotal!DRILL_TOTAL) And
QuerySumNotNULL(rsDrillSAT!TOTAL_SAT) Then
      If rsDrillTotal!DRILL_TOTAL Then
         aMeasures.AreaDrillAttend = rsDrillSAT!TOTAL_SAT /
rsDrillTotal!DRILL_TOTAL
      Else
         aMeasures.AreaDrillAttend = DefaultErrorValue
      End If
   Else
      aMeasures.AreaDrillAttend = DefaultErrorValue
   End If

   'Calculate Measure #8
   Call DisplayStatus(conMeasure8, MeasuresFrameIndex)

   Set rsNoASSN = dbAcropolis.OpenRecordset(NoASSNQuery, dbOpenSnapshot,
dbReadOnly)
   Call ElapsedTimer(Time$())

   Set rsNoLOSS = dbAcropolis.OpenRecordset(NoLOSSQuery, dbOpenSnapshot,
dbReadOnly)
   If QuerySumNotNULL(rsNoASSN!TOTAL_ASSN) Then
      If QuerySumNotNULL(rsNoLOSS!TOTAL_LOSS) And rsNoASSN!TOTAL_ASSN Then
         aMeasures.AreaLossRate = rsNoLOSS!TOTAL_LOSS / rsNoASSN!TOTAL_ASSN
      Else
         aMeasures.AreaLossRate = 0
      End If
```

```
    Else
       aMeasures.AreaLossRate = DefaultErrorValue
    End If

    'Calculate Measure #9
    Call DisplayStatus(conMeasure9, MeasuresFrameIndex)
    Set rsNoXFER = dbAcropolis.OpenRecordset(NoXFERQuery, dbOpenSnapshot,
dbReadOnly)

    If QuerySumNotNULL(rsNoASSN!TOTAL_ASSN) Then
       If QuerySumNotNULL(rsNoXFER!TOTAL_XFER) And rsNoASSN!TOTAL_ASSN Then
          aMeasures.AreaXferRate = rsNoXFER!TOTAL_XFER / rsNoASSN!TOTAL_ASSN
       Else
          aMeasures.AreaXferRate = 0
       End If
    Else
       aMeasures.AreaXferRate = DefaultErrorValue
    End If

    'Calculate Measure #10
    Call DisplayStatus(conMeasure10, MeasuresFrameIndex)

    If QuerySumNotNULL(rsNoASSN!TOTAL_ASSN) And
QuerySumNotNULL(rsNoREQD!TOTAL_REQD) Then
       aMeasures.AvgAreaMan = rsNoASSN!TOTAL_ASSN / rsNoREQD!TOTAL_REQD
    Else
       aMeasures.AvgAreaMan = 0
    End If

    'Calculate Measure #12
    Call DisplayStatus(conMeasure12, MeasuresFrameIndex)
    Set rsAvailClos = dbAcropolis.OpenRecordset(TotalAvailClosQuery,
dbOpenSnapshot, dbReadOnly)

    If QuerySumNotNULL(rsAvailClos!TOTAL_AVAIL) Then
       aMeasures.TotalAvailClos = rsAvailClos!TOTAL_AVAIL
    Else
       aMeasures.TotalAvailClos = DefaultErrorValue
    End If

    'Calculate Measure #13
    Call DisplayStatus(conMeasure13, MeasuresFrameIndex)
    Set rsIRR = dbAcropolis.OpenRecordset(TotalIRRQuery, dbOpenSnapshot,
dbReadOnly)

    If QueryCountNotNULL(rsIRR!TOTAL_IRR) Then
       aMeasures.IRR = rsIRR!TOTAL_IRR
    Else
       aMeasures.IRR = DefaultErrorValue
    End If

    'Calculate Measure #14
    Call DisplayStatus(conMeasure14, MeasuresFrameIndex)
    Set rsRecruitMarket = dbAcropolis.OpenRecordset(RecruitMarketQuery,
dbOpenSnapshot, dbReadOnly)

    If QuerySumNotNULL(rsRecruitMarket!TOTAL_MARKET) Then
       aMeasures.RecruitMarket = rsRecruitMarket!TOTAL_MARKET
    Else
       aMeasures.RecruitMarket = DefaultErrorValue
    End If

    'Calculate Measure #15
'    Call DisplayStatus(conMeasure15, MeasuresFrameIndex)
'    cQdefnSQL = ReassignQueryBegin & cMovUic.UIC & ReassignQueryEnd
'    Set rsReassignments = dbAcropolis.OpenRecordset(cQdefnSQL,
dbOpenSnapshot, dbReadOnly)
'
'    If QuerySumNotNULL(rsReassignments!TOTAL_RESERVISTS) Then
'       aMeasures.Reassignments = rsReassignments!TOTAL_RESERVISTS
```

```
'       Else
'          aMeasures.Reassignments = DefaultErrorValue
'       End If

'       'Calculate Measure #19
'       Call DisplayStatus(conMeasure19, MeasuresFrameIndex)
'       Set rsM0SAvailClos = dbAcropolis.0penRecordset(TotalClosM0S0uery,
    db0penSnapshot, dbRead0nly)
'
'       If 0uerySumNotNULL(rsM0SAvailClos!T0TAL_CL0S_M0S) Then
'          aMeasures.M0SAvailClos = rsM0SAvailClos!T0TAL_CL0S_M0S
'       Else
'          aMeasures.M0SAvailClos = DefaultErrorValue
'       End If

      'Calculate Measure #20
'       Call DisplayStatus(conMeasure20, MeasuresFrameIndex)
'       Set rsIRR_M0S = dbAcropolis.0penRecordset(TotalIRR_M0S0uery,
    db0penSnapshot, dbRead0nly)
'
'       If 0ueryCountNotNULL(rsIRR_M0S!T0TAL_IRR_M0S) Then
'          aMeasures.IRR_M0S = rsIRR_M0S!T0TAL_IRR_M0S
'       Else
'          aMeasures.IRR_M0S = DefaultErrorValue
'       End If
'       Call ElapsedTimer(Time$())

'    ElseIf Not aMeasures.M0vUIC_Match Then
      'Execute MapInfo Geo0uery via 0LE interface
'       Call DisplayStatus(aMeasures.FacID, MapinfoFrameIndex)
'
'       If bMap0ueryActive Then
'          Call MapInfo0uery(aMeasures.FacID)
'       Else
'          Call DisplayMessage(0LEfailedMSG, vbCritical, vb0K0nly, con0LEcaption)
'          bMapBasicRunning = False
'          Call cmdExitBtn_Click
'       End If
'       Call ElapsedTimer(Time$())

      'Build Temp Tables FinanceCY, AreaFacID, AreaZipCode, AreaUIC, AreaClosUIC
'       Call CreateAreaFacIDTempTable
'       Call CreateAreaUICTempTable
'       Call CreateAreaClosUICTempTable
'       Call CreateAreaZipCodeTempTable
'       Call CreateAreaG18M0STempTable
'       Call DisplayStatus(vbNullString, S0LFrameIndex)
'       iStatusBarMax = iStatusBarMax - 4
'       barS0Lstatus.Max = iStatusBarMax

      'Calculate Measure #19
'       Call DisplayStatus(conMeasure19, MeasuresFrameIndex)
'       Set rsM0SAvailClos = dbAcropolis.0penRecordset(TotalClosM0S0uery,
    db0penSnapshot, dbRead0nly)
'
'       If 0uerySumNotNULL(rsM0SAvailClos!T0TAL_CL0S_M0S) Then
'          aMeasures.M0SAvailClos = rsM0SAvailClos!T0TAL_CL0S_M0S
'       Else
'          aMeasures.M0SAvailClos = DefaultErrorValue
'       End If

      'Calculate Measure #20
'       Call DisplayStatus(conMeasure20, MeasuresFrameIndex)
'       Set rsIRR_M0S = dbAcropolis.0penRecordset(TotalIRR_M0S0uery,
    db0penSnapshot, dbRead0nly)
'
'       If 0ueryCountNotNULL(rsIRR_M0S!T0TAL_IRR_M0S) Then
'          aMeasures.IRR_M0S = rsIRR_M0S!T0TAL_IRR_M0S
'       Else
'          aMeasures.IRR_M0S = DefaultErrorValue
```

```
'       End If
'       Call ElapsedTimer(Time$())
'    End If

    'Delete Recordset Objects
    Set rsNoAuthNG = Nothing
    Set rsNoREQD = Nothing
    Set rsNoASSN = Nothing
    Set rsNoXFER = Nothing
    Set rsNoLOSS = Nothing
    Set rsDrillTotal = Nothing
    Set rsDrillSAT = Nothing
    Set rsAvailClos = Nothing
    Set rsRecruitMarket = Nothing
    Set rsReassignments = Nothing
    Set rsMOSAvailClos = Nothing
    Set rsIRR_MOS = Nothing
    Set rsIRR = Nothing
End Sub

Private Sub CloseGISQuery()
    cProcName = "CLOSE_GIS_QUERY"

    objGeoQuery.RunMenuCommand M_FILE_EXIT

    Call DisplayStatus(CompleteMSG, MeasuresFrameIndex)
    Call DisplayStatus(CompleteMSG, NonGISFrameIndex)
    Call DisplayStatus(vbNullString, MapinfoFrameIndex)
    Call DisplayStatus(CompleteMSG, MapinfoFrameIndex)
    Call DisplayStatus(CompleteMSG, GISNonSQLFrameIndex)
    Call DisplayStatus(CompleteMSG, SQLFrameIndex)

'    barSQLstatus.Value = iStatusBarMax

End Sub

Private Sub GetFacilityDistances(ByRef aMeasures As Measures_Type)
    cProcName = "GET_FACILITY_DISTANCES"

    Dim rsFacilityDistanceMeasures As Recordset  'Measures #11, #16, #17

On Error GoTo EH_FacDist

    'Open External Table
    Set rsFacilityDistanceMeasures =
    dbAcropolis.OpenRecordset(ExTbl_AreaDistance, dbOpenDynaset, dbReadOnly)

    Call DisplayStatus("AMSA Distance", GISNonSQLFrameIndex)
    Call DisplayStatus("ECS Distance", GISNonSQLFrameIndex)
    Call DisplayStatus("RZA Distance", GISNonSQLFrameIndex)

    If TableNotNULL(rsFacilityDistanceMeasures) Then
        With rsFacilityDistanceMeasures
            .MoveFirst
            'Input Facility Distances into Measure Table
                aMeasures.DistToAMSA = !AMSA_DIST    'Measure #16
                aMeasures.DistToECS = !ECS_DIST      'Measure #17
                aMeasures.DistToRecruit = !RZA_DIST  'Measure #11
        End With
    Else
        aMeasures.DistToAMSA = DefaultErrorValue
        aMeasures.DistToECS = DefaultErrorValue
        aMeasures.DistToRecruit = DefaultErrorValue
    End If
    Call DisplayStatus(vbNullString, GISNonSQLFrameIndex)

    'Delete Recordset Object
    rsFacilityDistanceMeasures.Close
    Set rsFacilityDistanceMeasures = Nothing
Exit Sub
```

```
EH_FacDist: 'Error Handler

   Select Case Err.Number
      Case ERR_ExtrnTableNotAttached
         'call AttachExternalTable (ExTbl_AreaDistance)
         Resume

      Case Else    'Trap all other Errors & Report
         Call OutputERROR_LOG
         Resume Next
   End Select
End Sub

Private Sub CreateFinanceTempTable()
   cProcName = "CREATE_FINANCE_TEMP_TABLES"
   Dim cQdefnSQL As String
   Dim dtDate As Date
   Dim cCY_QTR As String * 1

   'Initialize Date parameters
   dtDate = Now
   cCY_QTR = DatePart("q", dtDate)

   Select Case CInt(cCY_QTR)
     Case 4
        cQdefnSQL = Q1FY_SCRN
     Case 1
        cQdefnSQL = Q2FY_SCRN
     Case 2
        cQdefnSQL = Q3FY_SCRN
     Case 3
        cQdefnSQL = Q4FY_SCRN
   End Select

   'Create Interim Table [ FINANCE_CY ]
   cQdefnSQL = CY_FinanceQuery & cQdefnSQL
   Call DisplayStatus(Tbl_FinanceCY, SQLFrameIndex)
   Call BuildAccessTable(Tbl_FinanceCY, Qdef_FinanceCY, cQdefnSQL)

End Sub

Private Sub CreateAreaFacIDTempTable()
   cProcName = "CREATE_AREA_FACID_TEMP_TABLE"

   Dim cQdefnSQL As String

   'Create Interim Table [ AREA_FACID ]
   cQdefnSQL = AreaFacIDConvertQuery
   Call DisplayStatus(Tbl_AreaFacIDs, SQLFrameIndex)
   Call BuildAccessTable(Tbl_AreaFacIDs, Qdef_AreaFacIDs, cQdefnSQL)
End Sub

Private Sub CreateAreaUICTempTable()
   cProcName = "CREATE_AREA_UIC_TEMP_TABLE"

   Dim cQdefnSQL As String

   'Create Interim Table [ AREA_UIC ]
   cQdefnSQL = AreaUICQuery
   Call DisplayStatus(Tbl_AreaUICs, SQLFrameIndex)
   Call BuildAccessTable(Tbl_AreaUICs, Qdef_AreaUICs, cQdefnSQL)
End Sub

Private Sub CreateAreaClosUICTempTable()
   cProcName = "CREATE_AREA_UIC_TEMP_TABLE"

   Dim cQdefnSQL As String

   'Create Interim Table [ AREA_CLOS_UIC ]
```

```
    cQdefnSQL = AreaClosUICQuery
    Call DisplayStatus(Tbl_AreaClosUICs, SQLFrameIndex)
    Call BuildAccessTable(Tbl_AreaClosUICs, Qdef_AreaClosUICs, cQdefnSQL)
End Sub

Private Sub CreateAreaZipCodeTempTable()
    cProcName = "CREATE_AREA_UIC_TEMP_TABLE"

    Dim cQdefnSQL As String

    'Create Interim Table [ AREA_ZIPCODE ]
    cQdefnSQL = AreaZipCodeConvertQuery
    Call DisplayStatus(Tbl_AreaZipCodes, SQLFrameIndex)
    Call BuildAccessTable(Tbl_AreaZipCodes, Qdef_AreaZipCodes, cQdefnSQL)
End Sub

Private Sub CreateAreaG18ZipTempTable()
    cProcName = "CREATE_AREA_UIC_TEMP_TABLE"

    Dim cQdefnSQL As String

    'Create Interim Table [ AREA_G18_ZIP ]
    cQdefnSQL = AreaG18UICZipQuery
    Call DisplayStatus(Tbl_AreaG18UICZips, SQLFrameIndex)
    Call BuildAccessTable(Tbl_AreaG18UICZips, Qdef_AreaG18UICZips, cQdefnSQL)
End Sub

Private Sub CreateAreaG18M0STempTable()
    cProcName = "CREATE_AREA_UIC_TEMP_TABLE"

    Dim cQdefnSQL As String

    'Create Interim Table [ AREA_G18_M0S ]
    cQdefnSQL = AreaG18UICZipM0SQuery
    Call DisplayStatus(Tbl_AreaG18UICZipM0S, SQLFrameIndex)
    Call BuildAccessTable(Tbl_AreaG18UICZipM0S, Qdef_AreaG18UICZipM0S, cQdefnSQL)
    Call ElapsedTimer(Time$())
End Sub

Private Sub DeleteTempAccessTables()
    cProcName = "DELETE_TEMP_ACCESS_TABLES"

    Call DeleteAccessTable(Tbl_M0SInterest)
    Call DeleteAccessTable(Tbl_NoASSNxM0S)
    Call DeleteAccessTable(Tbl_FinanceCY)
    Call DeleteAccessTable(Tbl_AreaClosUICs)
    Call DeleteAccessTable(Tbl_AreaFacIDs)
    Call DeleteAccessTable(Tbl_AreaUICs)
    Call DeleteAccessTable(Tbl_AreaZipCodes)
    Call DeleteAccessTable(Tbl_AreaG18UICZips)
    Call DeleteAccessTable(Tbl_AreaG18UICZipM0S)
End Sub

Private Sub OutputComputedMeasures(ByRef aMeasures() As Measures_Type, _
    ByVal iFacIndex As Byte)
    cProcName = "OUTPUT_COMPUTED_MEASURES"
    Dim i As Byte

    'Output Computed Measures
    With AriesStatusBar
        .Panels(1).Text = OutputMeasuresMSG
        Call OutputMeasuresGrid(aMeasures(), iFacIndex)
        Call OutputAriesMeasuresExcel(aMeasures(), iFacIndex)
        DoEvents

        For i = 0 To iFacIndex
            Call OutputAriesFacilityInfo(aMeasures(i))
        Next i

        .Panels(1).Text = LDWLaunchMSG
```

```
      tabAriesMap.Tab = 2
      fraLDWcntrl.Visible = True
   End With
   Refresh
End Sub

Private Sub OutputAriesFacilityInfo(ByRef aMeasures As Measures_Type)
   cProcName = "OUTPUT_ARIES_MEASURES_TABLE"
   Dim i As Byte                          'Loop Variable
   Dim rsArchivedFacility As Recordset
   Dim QdefnSQL As String

   QdefnSQL = ArchivedFacilityQueryBegin & aMeasures.FacID &
   ArchivedFacilityQueryEnd
   AriesStatusBar.Panels(1).Text = OutputArchiveMSG
   Refresh

   'Open Facility Respository Database
   Set rsArchivedFacility = dbTeliko.OpenRecordset(Tbl_AriesArchive,
   dbOpenDynaset)

   If Not aMeasures.Archived Then
      Call ArchiveFacilityMeasures(rsArchivedFacility, aMeasures)

   ElseIf Not aMeasures.MovUIC_Match Then

      Call ArchiveFacilityMeasures(rsArchivedFacility, aMeasures)
   End If

   'Delete Database Objects
   Set rsArchivedFacility = Nothing
End Sub

Private Sub ArchiveFacilityMeasures(ByRef aMeasures As Measures_Type)

   With rsMeasures
      .AddNew
      'Transfer Measures to Proposed Facility Record
      !Fac_ID = aMeasures.FacID                'Measure #0
      !FAC_MAINT = aMeasures.FacBacklogdMaint  'Measure #1
      !FAC_OPCOST = aMeasures.OperatingCost    'Measure #2
      !FAC_AGE = aMeasures.FacAge              'Measure #3
      !FAC_COND = aMeasures.FacCond            'Measure #4
      !FAC_OWNED = aMeasures.FacOwned          'Measure #5
      !COMPETION = aMeasures.Competition       'Measure #6
      !DRILL_ATND = aMeasures.AreaDrillAttend  'Measure #7
      !AREA_LOSS = aMeasures.AreaLossRate      'Measure #8
      !AREA_XFER = aMeasures.AreaXferRate      'Measure #9
      !AVG_MANING = aMeasures.AvgAreaMan       'Measure #10
      !DIST_RZA = aMeasures.DistToRecruit      'Measure #11
      !AVAIL_CLOS = aMeasures.TotalAvailClos   'Measure #12
      !IRR = aMeasures.IRR                     'Measure #13
      !REC_MARKET = aMeasures.RecruitMarket    'Measure #14
'      !REASSIGN = aMeasures.Reassignments      'Measure #15
      !DIST_AMSA = aMeasures.DistToAMSA        'Measure #16
      !DIST_ECS = aMeasures.DistToECS          'Measure #17
      !WKND_USED = aMeasures.FacWkndUsed       'Measure #18
'      !MOS_AVAIL = aMeasures.MOSAvailClos      'Measure #19
'      !IRR_MOS = aMeasures.IRR_MOS             'Measure #20
'      !MovUIC = cMovUic.UIC

      'Save Changes to Database
      .Update
   End With
End Sub

Private Sub OutputMeasuresGrid(ByRef aMeasures() As Measures_Type, ByVal Index
   As Byte)
   cProcName = "OUTPUT_MEASURES_GRID"
   Dim i As Long
```

```
    Call InitMeasuresGrid(Index)

    With gridMeasures
       For i = 0 To Index
          .Col = i + 1
          .Row = 0

          .Text = aMeasures(i).FacID
          .Row = 1
          .Text = aMeasures(i).FacBacklogdMaint
          .Row = 2
          .Text = aMeasures(i).OperatingCost
          .Row = 3
          .Text = aMeasures(i).FacAge
          .Row = 4
          .Text = aMeasures(i).FacCond
          .Row = 5
          .Text = aMeasures(i).FacOwned
          .Row = 6
          .Text = aMeasures(i).Competition
          .Row = 7
          .Text = aMeasures(i).AreaDrillAttend
          .Row = 8
          .Text = aMeasures(i).AreaLossRate
          .Row = 9
          .Text = aMeasures(i).AreaXferRate
          .Row = 10
          .Text = aMeasures(i).AvgAreaMan
          .Row = 11
          .Text = aMeasures(i).DistToRecruit
          .Row = 12
          .Text = aMeasures(i).TotalAvailClos
          .Row = 13
          .Text = aMeasures(i).IRR
          .Row = 14
          .Text = aMeasures(i).RecruitMarket
          .Row = 15
          .Text = aMeasures(i).Reassignments
          .Row = 16
          .Text = aMeasures(i).DistToAMSA
          .Row = 17
          .Text = aMeasures(i).DistToECS
          .Row = 18
          .Text = aMeasures(i).FacWkndUsed
          .Row = 19
          .Text = aMeasures(i).MOSAvailClos
          .Row = 20
          .Text = aMeasures(i).IRR_MOS
       Next i
    End With
End Sub

Private Sub EnableLDWbuttons()
    cProcName = "ENABLE_LDW_BUTTONS"

    tbararies.Buttons("matrix").Enabled = True
    tbararies.Buttons("hierarchy").Enabled = True
    tbararies.Buttons("dynamic").Enabled = True
    tbararies.Buttons("stacked").Enabled = True
End Sub

Private Sub CompactAriesDataBase()
    cProcName = "COMPACT_ARIES_DATABASE"

    'Compact the Acropolis Database
    Set dbAcropolis = Nothing
    DBEngine.CompactDatabase App.Path & FILE_Acropolis, App.Path & FILE_Akron

    Kill App.Path & FILE_AcropolisBackup
```

```
    Name App.Path & FILE_Acropolis As App.Path & FILE_AcropolisBackup
    Name App.Path & FILE_Akron As App.Path & FILE_Acropolis

    'Compact the Repository Database
    Set dbTeliko = Nothing
    DBEngine.CompactDatabase App.Path & FILE_Teliko, App.Path & FILE_Titan

    Kill App.Path & FILE_TelikoBackup
    Name App.Path & FILE_Teliko As App.Path & FILE_TelikoBackup
    Name App.Path & FILE_Titan As App.Path & FILE_Teliko

    Call DisplayMessage(DatabaseCompactedMSG, vbInformation, vbOKOnly,
    conFileCaption)
End Sub
```

THIS PAGE LEFT INTENTIONALLY BLANK

Module 3. ARIES PUBLIC DECLARATIONS

Purpose: Initializes all public variables and constants; Contains the variables that represent the SQL business rules for calculating the ARU-DM decision parameters.

Source Type: Visual Basic for Applications

Source File: libDECLARATIONS.BAS

Code Listing:

```
Attribute VB_Name = "libDECLARATIONS"
Option Explicit

'=====================================================================
'     Aries Status Bar Messages
'=====================================================================
Public Const MovUICstatbarMSG = "Enter Moving Unit's (UIC)"

Public Const MovUIClengthMSG = "Moving UIC must be six " _
   & "Alpha-Numeric characters"

Public Const FacilityStatbarMSG = "Enter Facility Identifier " _
   & "for Proposed Facility "

Public Const FacIDlengthMSG = "Proposed Facility Identifier " _
   & "must be entered in { XX### } Format"

Public Const LaunchAriesMSG = "Select ARIES button to start " _
   & "Facility Comparisions"

Public Const SQLStatusMSG = "Calculating Measures for Facility : "
Public Const ClearButtonMSG = "Clears all User Entries"
Public Const ExitButtonMSG = "Ends Program"


'=====================================================================
'     SQL Status Display Messages
'=====================================================================
Public Const ArchivedFacilityMSG = "Checking ARIES Respository for " _
   & "Facilities..."
Public Const BuildMOSInterestMSG = "Creating MOS of Interest Tables..."
Public Const LaunchMapinfoMSG = "Loading Mapinfo and USARC Databases"
Public Const LDWLaunchMSG = "Ready to Launch LDW & Import " _
   & "Measures Table..."
Public Const OutputArchiveMSG = "Transferring Measures Table to " _
   & "Aries Archive..."
Public Const OutputMeasuresMSG = "Transferring Measures Table to " _
   & "EXCEL..."
Public Const LoadingArchiveMSG = "Retrieving Proposed Facility : "


'=====================================================================
'     Message Box User Messages
'=====================================================================
Public Const APIfailedMSG = "Win32 API Connection FAILED." _
      & vbCr & vbCr & "Unable to Complete Operation."

Public Const AriesConfirmMSG = "Conduct Data Analysis Comparisions" _
   & " of Proposed Facilities :"

Public Const DatabaseCompactedMSG = "ARIES Databases have been " _
```

```
   & "Compacted Successfully."

Public Const DuplicateFacilityMSG = "Unable to Conduct " _
   & "Analysis." & vbCr & vbCr & "DUPLICATE FACILITY PROPOSED"

Public Const FacIDinvalidMSG = "} is currently not on file in " _
   & "the COMMAND PLAN or USARC GEOREF File"

Public Const InvalidGISQueryMSG = "Unable to Conduct Analysis, " _
   & vbCr & vbCr & "Insufficient Amount of FACILITY(s) PROPOSED"

Public Const LDWresetMSG = "Do you wish to RESET the LDW Application?" _
   & vbCr & vbCr & "All Unsaved Data Changes will be Deleted."

Public Const MovUICinvalidMSG = "} is currently not on file in " _
   & "the COMMAND PLAN or USARC GEOREF File"

Public Const NoFacilityComparedMSG = "ARIES Facility Comparision " _
   & "Operation has not been Performed."

Public Const OLEfailedMSG = "OLE Connection FAILED." _
   & vbCr & vbCr & "Please RESTART Application."

Public Const CompleteMSG = "** COMPLETE **"
Public Const AreaG18MSG = "Area G18"
Public Const AreaIRRMSG = "Area IRR"
Public Const AreaQMAMSG = "Area QMA"
Public Const conNoAuthNG = "NO_AUTH_NG"
Public Const CompactDatabaseMSG = "Do you wish to COMPACT the Aries Databases?"
Public Const LDWactiveMSG = "LDW Application currently not Activated."
Public Const PrinterReadyMSG = "Ensure Printer is Ready...."
Public Const ProgressIndicatorMaxMSG = "Progress Bar Maximum Limit Exceeded"

Public Const conFacilityErrorCaption = "Proposed Facility Selection"
Public Const conUICcaption = "Unit Identification Code Selection"
Public Const conErrorCaption = "UNEXPECTED ERROR Trap"
Public Const conOLEcaption = "Windows 95 OLE Link Error"
Public Const conAPIcaption = "Logical Decisions-Win32 API Link"
Public Const conSQLcaption = "SQL Query Computation"
Public Const conFileCaption = "ARIES File Maintenance Operations"


'================================================================
'     SQL Query Definitions (Temporary Tables)
'================================================================
Public Const NoASSNxMOSQueryBegin = "Select MOS, Count(*) as " _
   & "MOS_COUNT Into NoASSNxMOS From G18Natl Where UIC = '"

Public Const NoASSNxMOSQueryEnd = "' Group by MOS Order by " _
   & "Count(*) DESC"

Public Const MOS_TotalQuery = "Select Sum(MOS_COUNT) as " _
   & "MOS_TOTAL From NoASSNxMOS"

Public Const MOS_Top3Query = "Select Top 3 MOS_COUNT From " _
   & "NoASSNxMOS"

Public Const MOS_INT_TotalQuery = "Select MOS Into " _
   & "MOS_INTEREST From NoASSNxMOS Order by MOS"

Public Const MOS_INT_Top3Query = "Select Top 3 MOS Into " _
   & "MOS_INTEREST From NoASSNXMOS Order By MOS"

Public Const CY_FinanceQuery = "Select UIC, Count(UIC) as UIC_TOTAL " _
   & "into FINANCE_CY From FINANCE_QTR Where "

Public Const Q1FY_SCRN = "(UTA1Q1PF + UTA2Q1PF + UTA3Q1PF + " _
   & "UTA4Q1PF) > 20 Group by UIC Order by UIC"

Public Const Q2FY_SCRN = "(UTA2Q1PF + UTA3Q1PF + UTA4Q1PF + " _
```

```
   & "UTA1QCFY) > 20 Group by UIC Order by UIC"

Public Const Q3FY_SCRN = "(UTA3Q1PF + UTA4Q1PF + UTA1QCFY + " _
   & "UTA2QCFY) > 20 Group by UIC Order by UIC"

Public Const Q4FY_SCRN = "(UTA4Q1PF + UTA1QCFY + UTA2QCFY + " _
   & "UTA3QCFY) > 20 Group by UIC Order by UIC"

Public Const AreaFacIDConvertQuery = "Select FAC_ID Into " _
   & "AREA_FACID From AreaFacID Order by FAC_ID"

Public Const AreaZipCodeConvertQuery = "Select Zip Into " _
   & "AREA_ZIPCODE From AreaZipCode Order by Zip"

Public Const AreaUICQuery = "Select Distinct UIC Into AREA_UIC " _
   & "From VALID_UIC Where VALID_UIC.FAC_ID = Any (Select " _
   & "AREA_FACID.FAC_ID From AREA_FACID) Order by UIC"

Public Const AreaClosUICQuery = "Select UIC Into AREA_CLOS_UIC " _
   & "From G17Natl Where Tier = '5B' and G17Natl.UIC = Any " _
   & "(Select AREA_UIC.UIC From AREA_UIC) Order by UIC"

Public Const AreaG18UICZipQuery = "Select Distinct UIC, ZipCode, " _
   & "Count(UIC) as UIC_TOTAL Into Area_G18_ZIP from G18 " _
   & "Group by UIC, ZipCode Order by UIC, ZipCode"

Public Const AreaG18UICZipMOSQuery = "Select Distinct UIC, ZipCode, " _
   & "MOS, Count(UIC) as UIC_TOTAL Into Area_G18_MOS from G18 " _
   & "Group by UIC, ZipCode, MOS Order by UIC, ZipCode, MOS"


'================================================================

'     SQL Query Definitions (Temporary Recordsets)
'================================================================
Public Const NoReqdQuery = "Select Sum(UIC_TOTAL) as TOTAL_REQD " _
   & "From G19Natl Where G19Natl.UIC = Any (Select AREA_UIC.UIC" _
   & " From AREA_UIC)"

Public Const DrillTotalQuery = "Select Sum(UIC_TOTAL) as DRILL_TOTAL " _
   & "From FINANCE_ Where FINANCE_.UIC = Any (Select AREA_UIC.UIC " _
   & "From AREA_UIC)"

Public Const DrillSATQuery = "Select Sum(UIC_TOTAL) as TOTAL_SAT " _
   & "From FINANCE_CY Where FINANCE_CY.UIC = Any (Select AREA_UIC.UIC " _
   & "From AREA_UIC)"

Public Const NoASSNQuery = "Select Sum(UIC_TOTAL) as TOTAL_ASSN " _
   & "From G18Natl_UIC Where G18Natl_UIC.UIC = Any " _
   & "(Select AREA_UIC.UIC From AREA_UIC)"

Public Const NoLOSSQuery = "Select Sum(UIC_TOTAL) as TOTAL_LOSS From " _
   & "FYxxLOSS Where FYxxLOSS.UIC = Any (Select AREA_UIC.UIC From " _
   & "AREA_UIC)"

Public Const NoXFERQuery = "Select Sum(UIC_TOTAL) as TOTAL_XFER From " _
   & "FYxxXFER Where FYxxXFER.UIC = Any (Select AREA_UIC.UIC From " _
   & "AREA_UIC)"

Public Const TotalAvailClosQuery = "Select Sum(UIC_TOTAL) as " _
   & "TOTAL_AVAIL From Area_G18_ZIP Where Area_G18_ZIP.UIC = " _
   & "Any (Select AREA_UIC.UIC From AREA_UIC) and " _
   & "Area_G18_ZIP.ZIPCODE = Any(Select AREA_ZIPCODE.ZIP From " _
   & "AREA_ZIPCODE)"

Public Const TotalIRRQuery = "Select Count(*) as TOTAL_IRR " _
   & "From IRR Where IRR.ZIP = Any (Select AREA_ZIPCODE.ZIP " _
   & "From AREA_ZIPCODE)"

Public Const RecruitMarketQuery = "Select Sum(MWCAT12+MWCAT3A+" _
```

```
    & "MBCAT12+MBCAT3A+MHCAT12+MHCAT3A) as TOTAL_MARKET From " _
    & "QMA Where QMA.ZIP = Any (Select AREA_ZIPCODE.ZIP " _
    & "From AREA_ZIPCODE)"

Public Const ReassignQueryBegin = "Select Sum(UIC_TOTAL) as " _
    & "TOTAL_RESERVISTS From Area_G18_ZIP Where " _
    & "Area_G18_ZIP.UIC = '"

Public Const ReassignQueryEnd = "' and Area_G18_ZIP.ZIPCODE = " _
    & "Any (Select AREA_ZIPCODE.ZIP From AREA_ZIPCODE)"

Public Const TotalClosMOSQuery = "Select Sum(UIC_TOTAL) as " _
    & "TOTAL_CLOS_MOS From Area_G18_MOS Where Area_G18_MOS.MOS " _
    & "= Any (Select MOS_INTEREST.MOS From MOS_INTEREST) and " _
    & "Area_G18_MOS.UIC = Any (Select AREA_CLOS_UIC.UIC From " _
    & "AREA_CLOS_UIC) and Area_G18_MOS.ZIPCODE = Any (Select " _
    & "AREA_ZIPCODE.ZIP From AREA_ZIPCODE)"

Public Const TotalIRR_MOSQuery = "Select Count(MOS) as " _
    & "TOTAL_IRR_MOS From IRR Where IRR.MOS = Any (Select " _
    & "MOS_INTEREST.MOS From MOS_INTEREST)"

Public Const ArchivedFacilityQueryBegin = "Select * From Repository Where FAC_ID
    = '"
Public Const ArchivedFacilityQueryEnd = "' Order by MOVUIC"

Public Const Qdef_AreaClosUICs = "Build AreaClosUIC"
Public Const Qdef_AreaUICs = "Build AreaUIC"
Public Const Qdef_AreaFacIDs = "Build AreaFacID"
Public Const Qdef_AreaG18UICZipMOS = "Build AreaG18MOS"
Public Const Qdef_AreaG18UICZips = "Build AreaG18Zip"
Public Const Qdef_AreaZipCodes = "Build AreaZipCode"
Public Const Qdef_DropTable = "Drop Table"
Public Const Qdef_FinanceCY = "Build FinanceCY"
Public Const Qdef_MOSInterest = "Build MOSInterest"
Public Const Qdef_NoASSNxMOS = "Build NoASSNxMOS"


'================================================================
'     Acropolis Database Table Names
'================================================================
Public Const Tbl_AreaClosUICs = "AREA_CLOS_UIC"
Public Const Tbl_AreaFacIDs = "AREA_FACID"
Public Const Tbl_AreaG18UICZipMOS = "AREA_G18_MOS"
Public Const Tbl_AreaG18UICZips = "AREA_G18_ZIP"
Public Const Tbl_AreaUICs = "AREA_UIC"
Public Const Tbl_AreaZipCodes = "AREA_ZIPCODE"
Public Const Tbl_AriesArchive = "REPOSITORY"
Public Const Tbl_CommandPlan = "CMDPLAN"
Public Const Tbl_Complex = "COMPLEX_"
Public Const Tbl_ErrorLog = "ERROR_LOG"
Public Const Tbl_FinanceCY = "FINANCE_CY"
Public Const Tbl_FPS = "FPS_"
Public Const Tbl_G17Natl = "G17Natl"
Public Const Tbl_Interest = "INTEREST_"
Public Const Tbl_MOSInterest = "MOS_INTEREST"
Public Const Tbl_NoASSNxMOS = "NoASSNxMOS"
Public Const Tbl_RPINFODT = "RPINFODT_"
Public Const Tbl_ValidateUIC = "VALID_UIC"
Public Const Tbl_ValidateUnit = "VALID_UNIT"

Public Const ExTbl_AreaDistance = "AreaDistance"
Public Const ExTbl_AreaFacIDs = "AreaFacID"
Public Const ExTbl_AreaUICs = "AreaUIC"
Public Const ExTbl_PropFacility = "PropFacID"


'================================================================
'     Aries OLE Class Names
'================================================================
```

```
Public Const Class_LDW = "OwlWindow"
Public Const Class_MapInfo = "xvt320mditask100"
Public Const Class_Excel = "XLMain"
Public Const Class_ARIES = "ThunderForm"
Public Const ARIES_Application = "A R I E S   v3.0 "


'====================================================================
'     Aries WinAPI32 Constants
'====================================================================
Public Const SW_Normal = 3
Public Const SW_Minimum = 2


'====================================================================
'     Aries Measure Table Names
'====================================================================
Public Const conMeasure0 = "Facility ID"
Public Const conMeasure1 = "Backlogged Maint"
Public Const conMeasure2 = "Operating Cost"
Public Const conMeasure3 = "Facility Age"
Public Const conMeasure4 = "Facility Condition"
Public Const conMeasure5 = "Facility Owned"
Public Const conMeasure6 = "Competition"
Public Const conMeasure7 = "Drill Attendance"
Public Const conMeasure8 = "Area Loss Rate"
Public Const conMeasure9 = "Area Xfer Rate"
Public Const conMeasure10 = "Avg Area Manning"
Public Const conMeasure11 = "RZA Distance"
Public Const conMeasure12 = "Total Avail (Close)"
Public Const conMeasure13 = "IRR"
Public Const conMeasure14 = "Recruit Market"
Public Const conMeasure15 = "Reassignments"
Public Const conMeasure16 = "AMSA Distance"
Public Const conMeasure17 = "ECS Distance"
Public Const conMeasure18 = "Weekend Usage"
Public Const conMeasure19 = "MOS Avail (Close)"
Public Const conMeasure20 = "IRR MOS"

Public Const conTotalMeasures = 20
Public Const conTotalInterimTables = 17
Public Const conSingleMessages = 3
Public Const conMaximumColumns = 4
Public Const conFiveColumnWidth = 967
Public Const conFourColumnWidth = 1210
Public Const conThreeColumnWidth = 1630
Public Const conTwoColumnWidth = 2460
Public Const conOneColumnWidth = 1370
Public Const MeasuresFrameIndex = 0
Public Const NonGISFrameIndex = 1
Public Const MapinfoFrameIndex = 2
Public Const GISNonSQLFrameIndex = 3
Public Const SQLFrameIndex = 4


'====================================================================
'     Aries Error Handler Codes
'====================================================================
Public Const ERR_ObjectExists = 3012
Public Const ERR_TableExists = 3010
Public Const ERR_TableNotExists = 3376
Public Const ERR_ExtrnTableNotAttached = 3078
Public Const ERR_FileNotFound = 53
Public Const ERR_LockTableFailure = 3211
Public Const ERR_ProgressBarMaxExceeded = 380
Public Const ERR_LDWRunning = 5
Public Const ERR_TooFewParameters = 3061
Public Const ERR_InvalidUseOFNull = 94
Public Const ERR_NoCurrentRecord = 3021
Public Const ERR_ReqdFileIndexMissing = 3015
```

```
Public Const ERR_ObjectUnloaded = 364
Public Const ERR_NotValidPath = 3044
Public Const ERR_DatabaseAlreadyOpen = 3356
Public Const ERR_ItemNotInCollection = 3265

Public Const DefaultErrorValue = -999


'=====================================================================
'     Aries File Paths
'=====================================================================
Public Const FILE_Akron = "\MS_Access\Akron.MDB"
Public Const FILE_Acropolis = "\MS_Access\Acropolis.MDB"
Public Const FILE_AcropolisBackup = "\MS_Access\Acropolis.BAK"
Public Const FILE_MajorCity = "\MapBasic\MapInfo\City_125.TAB"
Public Const FILE_EROS = "\MS_Access\Eros.MDB"
Public Const FILE_Excel = "\LDW\MS_Excel\AriesTAB.TXT"
Public Const FILE_GeoREF = "\MapBasic\UsarcData\Georef.TAB"
Public Const FILE_LDW = "C:\LDW\LDWstud.EXE"
Public Const FILE_LDW_Template = "\LDW\Template\NULL_TAB.LDW"
Public Const FILE_LDW_Excel = "\LDW\MS_Excel\AriesTAB.TXT~"
Public Const FILE_LDW_DefaultSave = "\LDW\Template\Default.LDW~"
Public Const FILE_LDW_ArchiveDir = "\LDW\Archive\"
Public Const FILE_MapBasic = "\MapBasic\AriesArch.MBX"
Public Const FILE_MapInfo = "C:\MAPINFO\MapInfoW.EXE"
Public Const FILE_MapWorkspace = "\MapBasic\AriesSQL.WOR"
Public Const FILE_States = "\MapBasic\MapInfo\States.TAB"
Public Const FILE_StateCapitals = "\MapBasic\MapInfo\Statecap.TAB"
Public Const FILE_Teliko = "\MS_Access\Teliko.MDB"
Public Const FILE_TelikoBackup = "\MS_Access\Teliko.BAK"
Public Const FILE_Titan = "\MS_Access\Titan.MDB"

Public Const DB_ACROPOLIS = "\MS_ACCESS\ACROPOLIS.mdb"
Public Const DB_ACROPOLISBAK = "\MS_ACCESS\ACROPOLIS.bak"
Public Const DB_MEASURES = "\MS_ACCESS\Measures.mdb"


'=====================================================================
'     Aries Measures (User Defined Type)
'=====================================================================
Type Measures_Type
   Archived As Boolean
   MovUIC_Match As Boolean
   FacID As String * 5
   FacBacklogdMaint As Single
   OperatingCost As Single
   FacAge As Integer
   FacCond As String * 5
   FacOwned As String * 1
   Competition As Integer
   AreaDrillAttend As Single
   AreaLossRate As Single
   AreaXferRate As Single
   AvgAreaMan As Single
   DistToRecruit As Single
   TotalAvailClos As Integer
   IRR As Integer
   RecruitMarket As Long
   Reassignments As Integer
   DistToAMSA As Single
   DistToECS As Single
   FacWkndUsed As Byte
   MOSAvailClos As Integer
   IRR_MOS As Integer
 End Type

Type MovingUnit_Type
   UIC As String * 6
   FacID As String * 5
   UnitName As String
```

```
   City As String
   State As String * 2
   ZipCode As String * 5
End Type

Type Facility_Type
   FacID As String * 5
   City As String
   State As String * 2
   ZipCode As String * 5
End Type

Type InfoTool_Type
    MapTable As String
    GeoFacID As String
End Type
```

THIS PAGE LEFT INTENTIONALLY BLANK

Module 4. ARIES PROCEDURES LIBRARY

Purpose: A library module that contains all the procedures called by other procedures more than one time.

Source Type: Visual Basic for Applications

Source File: LIBRARY.BAS

Code Listing:

```
Attribute VB_Name = "libPROCEDURES"

Declare Function GetClassName& Lib "user32" Alias "GetClassNameA" _
   (ByVal hwnd As Long, ByVal lpClassName As String, ByVal nMaxCount As Long)

Declare Function ShowWindow& Lib "user32" _
   (ByVal hwnd As Long, ByVal nCmdShow As Long)

Declare Function FindWindow& Lib "user32" Alias "FindWindowA" _
   (ByVal lpClassName As String, ByVal lpWindowName As String)

Public Function KeyUpperNumeric(nKeyVal As Integer) As Integer
   cProcName = "FUNC:KEY_UPPER_NUMERIC"

   'Passes Uppercase Letters
   'Converts Lowercase to Uppercase
   'Passes {0,1,2,3,4,5,6,7,8,9} & Cntrl Codes
   'REJECTS ALL else

   If nKeyVal > 26 Then
      If Chr(nKeyVal) >= "a" And Chr(nKeyVal) <= "z" Then
         nKeyVal = nKeyVal - 32
      End If

      If Chr(nKeyVal) >= "A" And Chr(nKeyVal) <= "Z" Then
         nKeyVal = nKeyVal
      Else
         nKeyVal = KeyNumeric(nKeyVal)
      End If
   End If

   KeyUpperNumeric = nKeyVal

End Function

Private Function KeyNumeric(nKeyVal As Integer) As Integer
   cProcName = "FUNC:KEY_NUMERIC"

   Dim cValidNumeric As String

   cValidNumeric = "0123456789"

   If nKeyVal > 26 Then
      If InStr(cValidNumeric, Chr(nKeyVal)) = 0 Then
         nKeyVal = 0
      End If
   End If

   KeyNumeric = nKeyVal
```

```
End Function

Public Function ValidateMovUIC(ByVal cUIC As String) As Boolean
   cProcName = "FUNC:VALIDATE_MOVUIC"

   Dim rsValidateMovUIC As Recordset

   'Develop Criteria for Search
   ValidateMovUIC = True

   'Open Validation Table
   Set rsValidateMovUIC = dbAcropolis.OpenRecordset(Tbl_ValidateUIC,
   dbOpenTable, dbReadOnly)
   With rsValidateMovUIC
      .Index = "UIC"
      .Seek "=", cUIC

      'Examine Table for first occurance of Moving Unit UIC
      If .NoMatch Then
         ValidateMovUIC = False
      Else                                  'Get Facility Information
         cMovUic.FacID = !Fac_ID
         cMovUic.UnitName = !UnitName
         cMovUic.City = !City
         cMovUic.State = !State
         cMovUic.ZipCode = !Zip
      End If
      .Close
   End With

   'Delete Recordset Object
   Set rsValidateMovUIC = Nothing

End Function

Public Function ValidateFacID(ByVal cFacID As String, _
   ByRef rsValidateFacID As Recordset) As Boolean
   cProcName = "FUNC:VALIDATE_FACID"

   'Develop Criteria for Search
   rsValidateFacID.Seek "=", cFacID

   'Examine Table for first occurance of FACID
   If rsValidateFacID.NoMatch Then
      ValidateFacID = False
   Else
      ValidateFacID = True
   End If

End Function

Public Function ComputeFacilityAge(ByVal DateAcquired As Date) As Integer
   cProcName = "FUNC:COMPUTE_FACILITY_AGE"
   Dim PresentDate As Date

   On Error Resume Next
   'Minimum Facility Age Value (Default)
   ComputeFacilityAge = 0

   'Today's Date
   PresentDate = Now

   If IsDate(DateAcquired) Then              'Valid Date
      ComputeFacilityAge = DateDiff("m", DateAcquired, PresentDate)
   End If

End Function

Public Function DisplayMessage(ByVal cMsgText As String, _
   ByVal iMsgIcon As Byte, ByVal iMsgBtn As Byte, _
```

```
    cMsgCaption As String) As Integer

    DisplayMessage = MsgBox(cMsgText, iMsgIcon + iMsgBtn, cMsgCaption)

End Function

Private Function FormatTime(ByVal iTime As Long) As String
    Dim iHour As Long
    Dim iMinute As Long
    Dim iSecond As Long
    Dim iRemainder As Long

    FormatTime = cQueryStartTime

    iHour = iTime \ 3600
    iRemainder = (iTime - (iHour * 3600))

    iMinute = iRemainder \ 60
    iSecond = iRemainder - (iMinute * 60)

    If Len(CStr(iHour)) = 2 Then
        FormatTime = CStr(iHour) & ":"
    Else
        FormatTime = "0" & CStr(iHour) & ":"
    End If

    If Len(CStr(iMinute)) = 2 Then
        FormatTime = FormatTime & CStr(iMinute) & ":"
    Else
        FormatTime = FormatTime & "0" & CStr(iMinute) & ":"
    End If

    If Len(CStr(iSecond)) = 2 Then
        FormatTime = FormatTime & CStr(iSecond)
    Else
        FormatTime = FormatTime & "0" & CStr(iSecond)
    End If

End Function

Public Function TableNotNULL(ByRef rsRecord As Recordset) As Boolean
    cProcName = "FUNC:TABLE_NOT_NULL"

    On Error Resume Next

    TableNotNULL = True

    If rsRecord.AbsolutePosition = -1 Then
        TableNotNULL = False
    End If

End Function

Public Function QuerySumNotNULL(ByVal cSumResult As Variant) As Boolean
    cProcName = "FUNC:QUERY_SUM_NOT_NULL"

    On Error Resume Next

    QuerySumNotNULL = True

    If IsNull(cSumResult) Then
        QuerySumNotNULL = False
    End If

End Function

Public Function QueryCountNotNULL(ByVal cSumResult As Variant) As Boolean
    cProcName = "FUNC:QUERY_COUNT_NOT_NULL"

    On Error Resume Next
```

```
    QueryCountNotNULL = True

    If cSumResult = 0 Then
       QueryCountNotNULL = False
    End If

End Function

Public Sub TrafficLight(ByVal Color As Long)
   cProcName = "TRAFFIC_LIGHT"

With frmAriesMain
   Select Case Color
      Case vbRed
         !shpRedLite.FillColor = Color
         !shpYellowLite.FillColor = vbBlack
         !shpGreenLite.FillColor = vbBlack
         !cmdAriesBtn.Enabled = False
      Case vbYellow
         !shpRedLite.FillColor = vbBlack
         !shpYellowLite.FillColor = Color
         !shpGreenLite.FillColor = vbBlack
         !cmdAriesBtn.Enabled = False
      Case vbGreen
         !shpRedLite.FillColor = vbBlack
         !shpYellowLite.FillColor = vbBlack
         !shpGreenLite.FillColor = Color
         !cmdAriesBtn.Enabled = True
      Case vbBlack
         !shpRedLite.FillColor = Color
         !shpYellowLite.FillColor = Color
         !shpGreenLite.FillColor = Color
         !cmdAriesBtn.Enabled = False
   End Select
End With

End Sub

Public Sub GetPropFacilityInfo(ByRef cFacilityInfo As Facility_Type, _
   ByRef rsFacilityData As Recordset)
   cProcName = "GET_FACILITY_INFO"
   Dim bValidFacility As Boolean

On Error GoTo EH_GetPropFacInfo

   bValidFacility = ValidateFacID(cFacilityInfo.FacID, rsFacilityData)

   If bValidFacility Then
      With cFacilityInfo
         .City = rsFacilityData!City
         .State = rsFacilityData!State
         .ZipCode = rsFacilityData!Zip
      End With

      Call LoadPropFacilityUICBox(cFacilityInfo.FacID)
   End If

Exit Sub

EH_GetPropFacInfo:

   Select Case Err.Number
      Case ERR_InvalidUse0FNull
         Resume Next

      Case Else   'Trap all other Errors & Report
         Call OutputERR0R_L0G
         Resume Next
   End Select
```

```
End Sub

Public Sub DisplayPropFacilityInfo(ByRef cFacility As Facility_Type)
   cProcName = "DISPLAY_FACILITY_INFO"

   With frmAriesMain
      !txtPropFacID.Text = cFacility.FacID
      !txtCity.Text = cFacility.City
      !txtState.Text = cFacility.State
      !txtZipCode.Text = cFacility.ZipCode
   End With

   Call DisplayPropFacilityUnitNames
End Sub

Public Sub LoadPropFacilityUICBox(ByRef cFacID As String)
   cProcName = "GET_PROPOSED_FACILITY_UIC_INFO"
   Dim i As Integer
   Dim rsFacilityUICs As Recordset

   Set rsFacilityUICs = dbAcropolis.OpenRecordset(Tbl_ValidateUIC, dbOpenTable,
   dbReadOnly)

   frmAriesMain!cboUIC.Clear
   With rsFacilityUICs
      .MoveFirst
      Do Until .EOF
         If cFacID = !Fac_ID Then
            frmAriesMain!cboUIC.AddItem !UIC
         End If
         .MoveNext
      Loop
      .Close
   End With

   'Test whether any Units are assigned
   With frmAriesMain
      If Not !cboUIC.ListCount = 0 Then
         !cboUIC.Text = !cboUIC.List(0)
      Else
         !cboUIC.Text = "-NONE-"
      End If
      .Refresh
   End With

   Set rsFacilityUICs = Nothing
End Sub

Public Sub DisplayPropFacilityUnitNames()
   cProcName = "DISPLAY_PROPOSED_FACILITY_UNITNAMES"
   Dim i As Integer
   Dim rsFacilityInfo As Recordset
   Dim rsFacilityUnit As Recordset

   Set rsFacilityInfo = dbAcropolis.OpenRecordset(Tbl_ValidateUIC, dbOpenTable,
   dbReadOnly)

   With frmAriesMain
      rsFacilityInfo.Index = "UIC"
      !lstbxUnitName.Clear

      If Not !cboUIC.ListCount = 0 Then
         For i = 0 To !cboUIC.ListCount - 1
            rsFacilityInfo.Seek "=", !cboUIC.List(i)
            If Not rsFacilityInfo.NoMatch Then
               !lstbxUnitName.AddItem rsFacilityInfo!UIC & vbTab _
                  & rsFacilityInfo!UnitName
            End If
         Next i
      Else
```

```
            !cboUIC.Text = "-NONE-"
            Set rsFacilityUnit = dbAcropolis.OpenRecordset(Tbl_ValidateUnit,
    dbOpenTable, dbReadOnly)
            rsFacilityUnit.Index = "FACID"
            rsFacilityUnit.Seek "=", !txtPropFacID.Text

            If rsFacilityUnit.NoMatch Then
               !lstbxUnitName.AddItem "** NOT AVAILABLE **"
            Else
               !lstbxUnitName.AddItem rsFacilityUnit!UnitName
            End If
            rsFacilityUnit.Close
         End If
         .Refresh
      End With

      rsFacilityInfo.Close
      Set rsFacilityInfo = Nothing
      Set rsFacilityUnit = Nothing
   End Sub

   Public Sub DisplayTreeViewInfo(ByRef tMapFacIDs() As InfoTool_Type, _
      ByVal Index As Long)
      cProcName = "DISPLAY_TREEVIEW_INFO"
      Dim rsFacilityUICs As Recordset
      Dim bUICExists As Boolean
      Dim nodeGIS As Node
      Dim i As Integer
      Dim j As Integer
      Dim cFacID As String * 5
      ReDim cKey(Index) As String * 5

      Set rsFacilityUICs = dbAcropolis.OpenRecordset(Tbl_ValidateUIC, dbOpenTable,
      dbReadOnly)

      With frmAriesMain
         !trvGISData.Enabled = True
         Set nodeGIS = !trvGISData.Nodes.Add( , , "ROOT", "USARC:GeoREF")

         For i = CInt(Index) To 1 Step -1
            cKey(i) = tMapFacIDs(i).GeoFacID
            cFacID = tMapFacIDs(i).GeoFacID
            Set nodeGIS = !trvGISData.Nodes.Add("ROOT", tvwChild, CStr(cKey(i)),
      cFacID)
            nodeGIS.Expanded = True

            bUICExists = False
            rsFacilityUICs.MoveFirst
            Do Until rsFacilityUICs.EOF
               If cFacID = rsFacilityUICs!Fac_ID Then
                  Set nodeGIS = !trvGISData.Nodes.Add(cKey(i), tvwChild, ,
      rsFacilityUICs!UIC)
                  nodeGIS.Expanded = True
                  bUICExists = True
               End If
               rsFacilityUICs.MoveNext
            Loop

            If Not bUICExists Then
               Set nodeGIS = !trvGISData.Nodes.Add(cKey(i), tvwChild, , "None")
               nodeGIS.Expanded = True
            End If
         Next i
         nodeGIS.EnsureVisible
         .Refresh

         rsFacilityUICs.Close
         Set rsFacilityUICs = Nothing
      End With
   End Sub
```

```
Public Sub LoadMovUIClistbox(ByRef cMovUic As MovingUnit_Type)
   cProcName = "LOAD_MOVUIC_LISTBOX"

   Dim iUnitNameLength As Byte
   Dim cUnitName As String

   cUnitName = cMovUic.UnitName
   iUnitNameLength = Len(cUnitName)

   With frmAriesMain!lstbxMovUIC
      .Clear
      .Enabled = True
      .AddItem "FacID:  " & cMovUic.FacID
      .AddItem "____________________"

      If iUnitNameLength > 18 Then
         .AddItem Left(cUnitName, 18)

         If iUnitNameLength > 36 Then
            .AddItem Mid(cUnitName, 19, 18)
            .AddItem Right(cUnitName, (iUnitNameLength - 36))
         Else
            .AddItem Right(cUnitName, (iUnitNameLength - 18))
         End If

      Else
         .AddItem cUnitName
      End If

      .AddItem "____________________"
      .AddItem cMovUic.City
      .AddItem cMovUic.State & ",  " & cMovUic.ZipCode
   End With
End Sub

Public Sub LoadFacilityListBox(ByVal Index As Integer, ByVal rsValidFacID _
   As Recordset, ByVal iUIC_Flag As Byte)
   cProcName = "LOAD_FACILITY_LISTBOX"

   Dim iUnitNameLength As Byte
   Dim cUnitName As String

   cUnitName = rsValidFacID!UnitName
   iUnitNameLength = Len(cUnitName)

   'Display Facility specific information
   With frmAriesMain!lstbxFacID(Index)
      .Clear
      .Enabled = True
      If iUIC_Flag = 0 Then
         .AddItem "UIC: " & rsValidFacID!UIC
      Else
         .AddItem "UIC: " & "-NONE-"
      End If
      .AddItem "____________________"

      'Parse Unitname to fit Listbox w/out needing scrollbars
      If iUnitNameLength > 18 Then
         .AddItem Left(cUnitName, 18)

         If iUnitNameLength > 36 Then
            .AddItem Mid(cUnitName, 19, 18)
            .AddItem Right(cUnitName, (iUnitNameLength - 36))
         Else
            .AddItem Right(cUnitName, (iUnitNameLength - 18))
         End If

      Else
         .AddItem cUnitName
```

```
      End If

      'Display City, State, Zip information
      .AddItem "_____"
      .AddItem rsValidFacID!City
      .AddItem rsValidFacID!State & ",  " & rsValidFacID!Zip
   End With
End Sub

Public Sub DisplayStatus(ByVal cMessage As String, Index As Byte)

   With frmAriesMain!fraStatus(Index)
      If Not .Enabled Then
         .Enabled = True
         .Refresh
      End If
   End With

   With frmAriesMain!lstbxStatus(Index)
      If Not (.Visible) Then
         .Visible = True
         .Refresh
      End If

      If Not (Len(frmAriesMain!txtStatus(Index)) = 0) Then
         .AddItem frmAriesMain!txtStatus(Index).Text
         .Refresh
      End If
   End With

   With frmAriesMain!txtStatus(Index)
      .Text = cMessage
      .Refresh
      If Not (cMessage = CompleteMSG) Then
         Call IncrementProgressBar
      Else
         .Enabled = False
         .Refresh
      End If
   End With

   Call ElapsedTimer(Time$())
End Sub

Public Sub IncrementProgressBar()

   On Error Resume Next

   iProgressIndicator = iProgressIndicator + 1
   frmAriesMain!barSQLstatus.Value = iProgressIndicator
   frmAriesMain!fraSQLstatus.Refresh
End Sub

Public Sub BuildAccessTable(ByVal cTableName As String, cQdefnName As String,
   cQdefText As String)
   cProcName = "BUILD_ACCESS_TABLE"

   Dim qdSQL As QueryDef

   On Error GoTo EH_BuildTable

   'Build SQL Query definition for Recordset Object
   Set qdSQL = dbAcropolis.CreateQueryDef(cQdefnName, cQdefText)
   dbAcropolis.Execute cQdefnName

   'Delete Query Definition & Table
   dbAcropolis.QueryDefs.Delete cQdefnName

Exit Sub
```

```
EH_BuildTable:  'Error Handler
   Select Case Err.Number
      Case ERR_ObjectExists   'Delete the Pre-Existing Query Definition
         dbAcropolis.QueryDefs.Delete cQdefnName
         Resume

      Case ERR_TableExists    'Delete the Pre-Existing Table
         Call DeleteAccessTable(cTableName)
         Resume

      Case Else   'Trap all other Errors & Report
         Call OutputERROR_LOG
         Resume Next
   End Select
End Sub

Public Sub DeleteAccessTable(ByVal cTableName As String)
   cProcName = "DELETE_ACCESS_TABLE"

   Dim qdDropTable As QueryDef
   Dim cDropTableDefn As String

   On Error GoTo EH_DeleteTable

   cDropTableDefn = "Drop Table " & cTableName

   Set qdDropTable = dbAcropolis.CreateQueryDef(Qdef_DropTable, cDropTableDefn)
   dbAcropolis.Execute Qdef_DropTable

   dbAcropolis.QueryDefs.Delete Qdef_DropTable

Exit Sub

EH_DeleteTable:
   Select Case Err.Number
      Case ERR_ObjectExists
         dbAcropolis.QueryDefs.Delete Qdef_DropTable
         Resume

      Case ERR_TableNotExists
         Resume Next

      Case Else
         Call OutputERROR_LOG
         Resume Next
   End Select
End Sub

Public Sub ElapsedTimer(ByVal cTime As String)
   Dim iHourNow As Long
   Dim iMinuteNow As Long
   Dim iSecondNow As Long

   Dim iStartHour As Long
   Dim iStartMinute As Long
   Dim iStartSecond As Long

   Dim iElapsedTime As Long
   Dim iStartTime As Long
   Dim iNowTime As Long

   Dim cElapsedTime As String * 8

   iHourNow = CLng(Left(cTime, 2))
   iMinuteNow = CLng(Mid(cTime, 4, 2))
   iSecondNow = CLng(Right(cTime, 2))

   iStartHour = CLng(Left(cQueryStartTime, 2))
   iStartMinute = CLng(Mid(cQueryStartTime, 4, 2))
   iStartSecond = CLng(Right(cQueryStartTime, 2))
```

```
    iStartTime = (iStartHour * 3600) + (iStartMinute * 60) + iStartSecond
    iNowTime = (iHourNow * 3600) + (iMinuteNow * 60) + iSecondNow

    iElapsedTime = iNowTime - iStartTime

    cElapsedTime = FormatTime(iElapsedTime)

    With frmAriesMain
       !AriesStatusBar.Panels(3).Text = "Elapsed " & cElapsedTime
       .Refresh
    End With
End Sub

Public Sub TimeDelay(ByVal cTime As String, _
    ByVal iTimeDelay As Integer)
    cProcName = "TIME_DELAY"
    Dim iSecond As Integer
    Dim iMinute As Integer
    Dim iNowMinute As Integer
    Dim iNowSecond As Integer

    Dim iExitTime As Integer
    Dim bRolloverFlag As Boolean

    bRolloverFlag = False
    iSecond = CInt(Right(cTime, 2))
    iMinute = CInt(Mid(cTime, 4, 2)) + 1

    iExitTime = iSecond + iTimeDelay
    If iExitTime > 59 Then
       iExitTime = iExitTime - 59
       bRolloverFlag = True
    End If

    iNowSecond = CInt(Right(Time$(), 2))
    If bRolloverFlag Then
       iNowMinute = CInt(Mid(Time$(), 4, 2))
       Do
          iNowMinute = CInt(Mid(Time$(), 4, 2))
          iNowSecond = CInt(Right(Time$(), 2))
       Loop While iNowMinute < iMinute
    Else
       Do
          iNowSecond = CInt(Right(Time$(), 2))
       Loop While iNowSecond < iExitTime
    End If
End Sub
```

Module 5. OLE PROCEDURES LIBRARY

Purpose: A library module that contains all the procedures supporting OLE communications between COTS components.

Source Type: Visual Basic for Applications

Source File: OLE_LIBRARY.BAS

Code Listing:

```
Attribute VB_Name = "libOLE"
Option Explicit

Public Function ConnectOLEobject(ByRef objOLE As Object, ByVal cOLEobjectName
    As String) As Boolean
    cProcName = "FUNC:OPEN_OLE_OBJECT"

On Error Resume Next

    'Set Boolean Flags
    ConnectOLEobject = True
    Set objOLE = Nothing

    'Initiate a new instance
    Set objOLE = CreateObject(cOLEobjectName)

    'Verify OLE connection established
    If objOLE Is Nothing Then        'OLE Error
        Call DisplayMessage(OLEfailedMSG, vbExclamation, vbOK, conOLEcaption)
        ConnectOLEobject = False
    Else
        objOLE.Visible = False
    End If

    DoEvents

End Function

Public Function OpenLDWobject()
    cProcName = "FUNC:OPEN_LDW_OBJECT"
    Dim cFilePath As String

On Error GoTo EH_OpenLDW

    OpenLDWobject = True
    cFilePath = FILE_LDW
    'Determine if an Instance of LDW already running
    iLDWwinID = FindWindow(Class_LDW, vbNullString)

    If iLDWwinID Then
        Call ResetLDWobject
    Else
        iLDWwinID = Shell(cFilePath, vbNormalFocus)
        frmAriesMain!mnuFCloseLDW.Enabled = True
        'Clear LDW Splash Screen
        SendKeys "~", True
    End If
Exit Function

EH_OpenLDW:                      'LDW Failed to Load properly.
    OpenLDWobject = False
```

```
End Function

Public Function VerifyLDWobjectConnection(ByVal cMessageText As String) As
    Boolean
   Dim iLDW_hWnd As Long

   iLDW_hWnd = FindWindow(Class_LDW, vbNullString)
   If iLDW_hWnd = 0 Then
      VerifyLDWobjectConnection = False
      Call DisplayMessage(cMessageText, vbCritical, vb0K0nly, conAPIcaption)
   Else
      VerifyLDWobjectConnection = True
      ShowWindow iLDW_hWnd, SW_Minimum
      ShowWindow iLDW_hWnd, SW_Normal
   End If
End Function

Public Sub InitializeMapInfoMenu()
   cProcName = "INITIALIZE_MAP_MENU"

   objGeoSelect.Do "Create Menu ""MapperShortcut"" ID 17 As ""(-"" "

End Sub

Public Sub InitializeMapInfoToolbar()
   cProcName = "INITIALIZE_MAP_T00L_BAR"

   objGeoSelect.Do "Create ButtonPad ""InfoTool"" As " & _
      "ToolButton ID 501 DrawMode 34 Calling 0LE ""ProcessInfoToolButton"""

End Sub

Public Sub InitializeMapInfoStatusBar()
   cProcName = "INITIALIZE_MAP_STATUS_BAR"

   objGeoSelect.Do "Set Map Display Zoom"
   objGeoSelect.Do "Alter Menu Item ID 1101 Check"
   objGeoSelect.Do "Alter Menu Item ID 1102 UnCheck"
   objGeoSelect.Do "Alter Menu Item ID 1103 UnCheck"

End Sub

Public Sub 0penUS_StatesLayer()
   cProcName = "DISPLAY_US_STATES_LAYER"

   Dim cFilePath As String

   cFilePath = Chr(34) & App.Path & FILE_States & Chr(34)
   objGeoSelect.Do "0pen Table " & cFilePath & " Interactive"

End Sub

Public Sub 0penUSARC_GeoRefLayer()
   cProcName = "DISPLAY_USARC_GE0REF_LAYER"

   Dim cFilePath As String

   cFilePath = Chr(34) & App.Path & FILE_GeoREF & Chr(34)
   objGeoSelect.Do "0pen Table " & cFilePath & " Interactive"

End Sub

Public Sub 0penStateCapitalsLayer()
   cProcName = "DISPLAY_STATE_CAPITALS_LAYER"

   Dim cFilePath As String

   cFilePath = Chr(34) & App.Path & FILE_StateCapitals & Chr(34)
   objGeoSelect.Do "0pen Table " & cFilePath & " Interactive"
```

```
End Sub

Public Sub OpenMajorCityLayer()
   cProcName = "DISPLAY_MAJOR_CITIES_LAYER"

   Dim cFilePath As String

   cFilePath = Chr(34) & App.Path & FILE_MajorCity & Chr(34)
   objGeoSelect.Do "Open Table " & cFilePath & " Interactive"

End Sub


Public Sub DisplayMapLabels()
   cProcName = "DISPLAY_MAP_LABELS"

   objGeoSelect.Do "Map From Statecap, City_125, Georef, States"
   iMapInfoWinID = CLng(objGeoSelect.Eval("FrontWindow()"))
   objGeoSelect.Do "Set Map Redraw Off"

   'State Capitals layer Labels & Symbols
   objGeoSelect.Do "Set Map Layer ""Statecap"" Display Global Selectable Off "
   objGeoSelect.Do "Set Map Layer ""Statecap"" Label Auto On Overlap On " & _
      "Duplicates On  With Proper$(Capital) "
   objGeoSelect.Do "Set Map Layer ""Statecap"" Label Line Simple " _
      & "Position Above Offset 3 Font (""Arial"", 803,8,8388608,16777215) " _
      & "Global Symbol (35, 16711680, 18) "
   objGeoSelect.Do "Set Map Layer ""Statecap"" Label Visibility Zoom (400,
   2000)" _
      & " Units ""mi"""

   'Major City layer Labels & Symbols
   objGeoSelect.Do "Set Map Layer ""City_125"" Display Global Selectable Off "
   objGeoSelect.Do "Set Map Layer ""City_125"" Label Auto On Overlap On " & _
      "Duplicates On  With Proper$(City) Zoom (0, 3000) Units ""mi"" On "
   objGeoSelect.Do "Set Map Layer ""City_125"" Label Line Simple " _
      & "Position Above Offset 3 Font (""Arial"", 291,10,16776960,0) " _
      & "Global Symbol (46, 16776960, 12) "
   objGeoSelect.Do "Set Map Layer ""City_125"" Label Visibility Zoom (250,
   1000)" _
      & " Units ""mi"""

   'GeoREF layer Labels & Symbols
   objGeoSelect.Do "Set Map Layer ""Georef"" Display Global Selectable On "
   objGeoSelect.Do "Set Map Layer ""Georef"" Label Auto On Overlap On " & _
      "Duplicates Off  With Proper$(Fac_city) Zoom (0, 500) Units ""mi"" On "
   objGeoSelect.Do "Set Map Layer ""Georef"" Label Line Simple " _
      & "Position Above Right Offset 3 Font (""Arial"", 803,10,32768,16777215)"

   _
      & " Pen (2,14,32768) Global Symbol (45, 32768, 18) "
   objGeoSelect.Do "Set Map Layer ""Georef"" Label Visibility Zoom (0, 700) " _
      & "Units ""mi"""

   'States layer Labels
   objGeoSelect.Do "Set Map Layer ""States"" Selectable Off Label Line Simple "
   objGeoSelect.Do "Set Map Layer ""States"" Label Auto On Overlap On " _
      & "Duplicates On Offset 0 "
   objGeoSelect.Do "Set Map Layer ""States"" Label With Proper$(State) "
   objGeoSelect.Do "Set Map Layer ""States"" Label Font (""Bookman Old Style"","

   _
      & "1827,18,0,16777215) "
   objGeoSelect.Do "Set Map Layer ""States"" Label Visibility Zoom (100, 2000) "

   _
      & "Units ""mi"""

   objGeoSelect.Do "Set Map Redraw On"
End Sub

Public Sub PositionMap()
```

```
    cProcName = "POSITION_MAP"

    objGeoSelect.Do "Map From Statecap, City_125, Georef, States"
    objGeoSelect.Do "Set Map Center (-96,37) Zoom 3500 Units ""mi"""
    objGeoSelect.Do "Set Map Layer 0 Editable On"

End Sub

Public Sub MapInfoQuery(ByVal cFacID As String)
    cProcName = "MAPINFO_QUERY"

    Dim i As Byte                          'Loop Variable
    Dim cFilePath As String
    Dim rsProposedFacility As Recordset

    Set rsProposedFacility = dbAcropolis.OpenRecordset(ExTbl_PropFacility,
    dbOpenDynaset)
    With rsProposedFacility

    'Delete Previous Proposed Facilities
       Do Until .EOF
          .Delete
          .MoveNext
       Loop
       'Output Proposed Facility ID to External Database
       .AddNew
       !Fac_ID = cFacID
       .Update
       .Close
    End With
    Set rsProposedFacility = Nothing

    objGeoQuery.Visible = True
    If Not bMapBasicRunning Then
       cFilePath = Chr(34) & App.Path & FILE_MapWorkspace & Chr(34)
       objGeoQuery.Do "Run Application " & cFilePath
    End If

    cFilePath = Chr(34) & App.Path & FILE_MapBasic & Chr(34)
    frmAriesMain!tabAriesMap.Tab = 1

    objGeoQuery.Do "Run Application " & cFilePath
    bMapBasicRunning = True

    Call DisplayStatus(AreaGISMSG, GISNonSQLFrameIndex)
    Call DisplayStatus(AreaIRRMSG, GISNonSQLFrameIndex)
    Call DisplayStatus(AreaQMAMSG, GISNonSQLFrameIndex)

End Sub

Public Sub OutputAriesMeasuresExcel(ByRef aMeasures() As Measures_Type, ByVal
    iFacIndex As Byte)
    cProcName = "OUTPUT_ARIES_MEASURES_EXCEL"

    Dim objWorkSht As Object
    Dim cFilePath As String

    Dim i As Byte
    Dim j As Byte

On Error GoTo EH_ExcelMeasures

    Screen.MousePointer = vbHourglass

    If ConnectOLEobject(objExcel, "Excel.Application") Then
       objExcel.Visible = True
       objExcel.Workbooks.Add

       Set objWorkSht = objExcel.ActiveSheet
       With objWorkSht
```

```
.cells(1, 1).Value = "ALTERNATIVES"
.cells(2, 1).Value = "NAME"
.cells(2, 2).Value = "NUMBER"
.cells(2, 3).Value = "NUMBER"
.cells(2, 4).Value = "NUMBER"
.cells(2, 5).Value = "NUMBER"
.cells(2, 6).Value = "NUMBER"
.cells(2, 7).Value = "LABEL"
.cells(2, 8).Value = "NUMBER"
.cells(2, 9).Value = "NUMBER"
.cells(2, 10).Value = "NUMBER"
.cells(2, 11).Value = "NUMBER"
.cells(2, 12).Value = "NUMBER"
.cells(2, 13).Value = "NUMBER"
.cells(2, 14).Value = "NUMBER"
.cells(2, 15).Value = "NUMBER"
.cells(2, 16).Value = "NUMBER"
.cells(2, 17).Value = "NUMBER"
.cells(2, 17).Value = "LABEL"
.cells(2, 18).Value = "NUMBER"
.cells(2, 19).Value = "NUMBER"
.cells(2, 20).Value = "NUMBER"
.cells(2, 21).Value = "NUMBER"

'Output Measure Titles for LDW Import
.cells(3, 1).Value = "NAME"
.cells(3, 2).Value = "Closing_Unit_Xfers"
.cells(3, 3).Value = "Avail_M0S_ClosUnits"
.cells(3, 4).Value = "IRR_Available"
.cells(3, 5).Value = "Avail_M0S_IRR"
.cells(3, 6).Value = "Fac_Age"
.cells(3, 7).Value = "Fac_Owned"
.cells(3, 8).Value = "Fac_Operating_Costs"
.cells(3, 9).Value = "Dist_to_AMSA"
.cells(3, 10).Value = "Dist_to_ECS"
.cells(3, 11).Value = "Area_Transfer_Rate"
.cells(3, 12).Value = "Avg_Area_Manning"
.cells(3, 13).Value = "Area_Drill_Attendnc"
.cells(3, 14).Value = "Dist-to-Recruiter"
.cells(3, 15).Value = "Recruit_Market"
.cells(3, 16).Value = "Reassignments"
.cells(3, 17).Value = "Fac_Condition"
.cells(3, 18).Value = "Competition"
.cells(3, 19).Value = "Fac_Weekend_Use"
.cells(3, 20).Value = "Fac_Backlogd_Maint"
.cells(3, 21).Value = "Area_Loss_Rate"

DoEvents
For i = 0 To iFacIndex
   j = i + 4

   .cells(j, 1).Value = "'" & aMeasures(i).FacID
   .cells(j, 2).Value = aMeasures(i).TotalAvailClos
   .cells(j, 3).Value = aMeasures(i).M0SAvailClos
   .cells(j, 4).Value = aMeasures(i).IRR
   .cells(j, 5).Value = aMeasures(i).IRR_M0S
   .cells(j, 6).Value = aMeasures(i).FacAge
   .cells(j, 7).Value = "'" & aMeasures(i).FacOwned
   .cells(j, 8).Value = aMeasures(i).OperatingCost
   .cells(j, 9).Value = aMeasures(i).DistToAMSA
   .cells(j, 10).Value = aMeasures(i).DistToECS
   .cells(j, 11).Value = aMeasures(i).AreaXferRate
   .cells(j, 12).Value = aMeasures(i).AvgAreaMan
   .cells(j, 13).Value = aMeasures(i).AreaDrillAttend
   .cells(j, 14).Value = aMeasures(i).DistToRecruit
   .cells(j, 15).Value = aMeasures(i).RecruitMarket
   .cells(j, 16).Value = aMeasures(i).Reassignments
   .cells(j, 17).Value = "'" & aMeasures(i).FacCond
   .cells(j, 18).Value = aMeasures(i).Competition
   .cells(j, 19).Value = aMeasures(i).FacWkndUsed
```

```
            .cells(j, 20).Value = aMeasures(i).FacBacklogdMaint
            .cells(j, 21).Value = aMeasures(i).AreaLossRate

            DoEvents
         Next i

        .cells(4, 1).Value = "_MOVING UNIT"

      End With

      cFilePath = App.Path & FILE_Excel
      SendKeys "%FA", True
      SendKeys cFilePath, True

      SendKeys "%t", True
      SendKeys "{down}{down}{down}", True
      SendKeys "%S", True
      SendKeys "%Y~", True

      If Not objExcel Is Nothing Then
         objExcel.ActiveWorkBook.Close False
      End If

      objExcel.Quit
      Set objExcel = Nothing
   End If
   Screen.MousePointer = vbDefault

   Exit Sub

EH_ExcelMeasures:
   Call OutputERROR_LOG
   Set objWorkSht = Nothing
End Sub

Public Sub ResetLDWobject()
   cProcName = "RESET_LDW_OBJECT"

   If VerifyLDWobjectConnection(APIfailedMSG) Then
      SendKeys "%FN", True
      SendKeys "%N", True
      DoEvents
   End If
End Sub

Public Sub ImportLDWdata()
   Dim cFilePath As String
   Dim iLDW_hWnd As Long

   If bLDWactive Then
      VerifyLDWobjectConnection (APIfailedMSG)

      'Load LDW Template File
      cFilePath = App.Path & FILE_LDW_Template
      SendKeys "%FO", True
      SendKeys cFilePath, True
      SendKeys "{ENTER}{ENTER}", True
      DoEvents

      'Import Excel Tab delimited file
      cFilePath = App.Path & FILE_LDW_Excel
      SendKeys "%FI", True
      SendKeys "{tab}{tab}{down}{tab}{down}{down}{down}{down}~", True
      SendKeys cFilePath, True
      SendKeys "%Y%S%N%O", True
      DoEvents

      Call SendCommentsLDW

      'Save Temp file of Imported measure comparisions
```

```
      cFilePath = App.Path & FILE_LDW_DefaultSave
      SendKeys "%FA", True
      SendKeys cFilePath, True
      SendKeys "%Y", True
      DoEvents
   End If
End Sub

Public Sub SendCommentsLDW()
   Dim rsFacilityInfo As Recordset
   Dim i As Byte
   Dim j As Byte
   Dim cFacID As String * 5
   Dim cCity As String
   Dim cState As String * 2

   On Error GoTo EH_Comment

   Set rsFacilityInfo = dbAcropolis.OpenRecordset(Tbl_ValidateUnit, dbOpenTable,
   dbReadOnly)

   With rsFacilityInfo
      .Index = "FACID"

      For i = 0 To 4
         .Seek "=", aFacID(i)
         If Not .NoMatch Then
            cFacID = !Fac_ID
            cCity = !City
            cState = !State
            'Send location information to LDW
            SendKeys "%EM~", True
            For j = 0 To i
               SendKeys "{down}", True
            Next j
            SendKeys "~{Tab}{Tab}{Tab}", True
            SendKeys cFacID, True
            SendKeys "~", True
            SendKeys cCity & ",  " & cState, True
            SendKeys "{tab}{tab}~"
         End If
      Next i
   End With
   Set rsFacilityInfo = Nothing

   Exit Sub

EH_Comment:
   Exit Sub
End Sub

Public Sub ViewLDWmatrix()
   Dim iLDW_hWnd As Long


   If bLDWactive Then
      VerifyLDWobjectConnection (APIfailedMSG)
      DoEvents
      SendKeys "%VM~", True
   End If
End Sub

Public Sub ActivateLDWReports()

   'Print ARIES Model Hierarchy w/ Global weights
   SendKeys "%W2", True
   SendKeys "%H{down}G", True
   Call TimeDelay(Time$(), 2)
   SendKeys "%FP", True
   Call TimeDelay(Time$(), 2)      ' 2 second delay
```

```
    SendKeys "~~", True

    Call TimeDelay(Time$(), 13)     '13 Second delay

    'Print ARIES Model Stacked bar ranking
    SendKeys "%SB", True
    Call TimeDelay(Time$(), 2)      '2 Second delay
    SendKeys "~", True
    SendKeys "%FP~", True

    Call TimeDelay(Time$(), 5)      '5 Second delay

    'Print ARIES Model Comment Summary
    SendKeys "%R00~", True
    SendKeys "%FP", True

    Call TimeDelay(Time$(), 5)      '5 Second delay

    'Change Printer Setup for Landscape Mode
    SendKeys "%FR%L~", True
    Call TimeDelay(Time$(), 1)      '1 Second delay

    'Print ARIES Model Preference Set Summary
    SendKeys "%SP~", True
    SendKeys "%FP", True

    Call TimeDelay(Time$(), 5)      '5 Second delay

    'Print ARIES Model Ranking results matrix
    SendKeys "%SMM~", True
    SendKeys "%FP", True

    Call TimeDelay(Time$(), 5)      '5 Second delay

    'Change Printer Setup for Portrait Mode
    SendKeys "%FR%R~", True
    Call TimeDelay(Time$(), 1)      '1 Second delay
End Sub
```

Module 6. OLE OBJECT CLASS DEFINITION

Purpose: An object class description necessary for communicating with the integrated MapInfo component.

Source Type: Visual Basic for Applications

Source File: OLE_CALLBACK.CLS

Code Listing:

```
VERSION 1.0 CLASS

BEGIN
   MultiUse = -1  'True
END

Attribute VB_Name = "cls0LECallback"
   Attribute VB_Creatable = False
   Attribute VB_Exposed = True
   Option Explicit

Private Function GetField(ByVal theStr As String, ByVal delimiter As String, _
   ByVal whichField As Integer) As String
   cProcName = "FUNC:GET_FIELD"

   Dim i As Integer
   Dim startPos As Integer
   Dim argCount As Integer
   Dim tmpStr, result As String

'* if the input string was null, or they asked for a field # less than 1,
'* return an empty string as the result
   If Len(theStr) = 0 Or whichField < 1 Then
      GetField = ""
      Exit Function
   End If

'* loop through til we get the field we want. If the string contains fewer
'* fields than the field number requested, a zero-length string is returned.
   argCount = 0
   tmpStr = theStr

   Do
      argCount = argCount + 1
      result = Left$(tmpStr, InStr(tmpStr, delimiter) - 1)
      tmpStr = Mid$(tmpStr, InStr(tmpStr, delimiter) + 1)
   Loop While InStr(tmpStr, delimiter) > 0 And argCount < whichField

'* if we dropped out of the while loop because InStr() returned 0, and
'* we haven't gotten to the desired field yet, take one more step in case
'* we are actually looking for the last field in the string.
   If argCount < whichField Then
      argCount = argCount + 1
      result = tmpStr
   End If

'* if we found our field, argCount will equal whichField. return the
'* result string, otherwise, return a zero-length string
   If argCount = whichField Then
      GetField = result
   Else
```

```
      GetField = ""
   End If

End Function
'*************************************************************************
'   ProcessInfoToolButton
'       This method is called when the custom tools defined in MapInfo
'       in the "InitializeMapInfoConnection" are used in the map window.
'       This method determines which tool was used ( a point select tool,
'       and a rectangle select tool ), and performs the appropriate action
'*************************************************************************

Public Sub ProcessInfoToolButton(ByVal CommandInfoText As String)
  cProcName = "PROCESS_TOOL_BUTTON"

   Dim iSelectedButton As Integer
   Dim fXpt As Double
   Dim fYpt As Double
   Dim nHits As Long

'Verify that the Windows Message came from MapInfo
'If it did come from MapInfo, strip off the "MI:" prefix:

   If Left$(CommandInfoText, 3) = "MI:" Then
      CommandInfoText = Mid$(CommandInfoText, 4, 9999)
   Else
      Exit Sub
   End If      'Determine which Tool Selected

      iSelectedButton = CInt(GetField(CommandInfoText, ",", CMD_INFO_TOOLBTN))
      Select Case iSelectedButton
         Case 501       'info tool
            fXpt = CDbl(GetField(CommandInfoText, ",", CMD_INFO_X))     '* get x
   and y
            fYpt = CDbl(GetField(CommandInfoText, ",", CMD_INFO_Y))     '* from
   CommandInfo

         Call DisplaySelectedFacility(fXpt, fYpt)
      End Select

End Sub

Private Sub DisplaySelectedFacility(ByVal fXpt As Double, ByVal fYpt As
   Double)
   cProcName = "DISPLAY_SELECTED_FACILITY"

   Dim nHits As Long
   Dim iRecNum As Long
   Dim i As Long
   Dim tMapHits() As InfoTool_Type
   Dim tFacilityInfo As Facility_Type
   Dim rsGeoREFdata As Recordset

   'Get number of "Hits" using the MapBasic function {SearchPoint()}
   'FORMAT: nHits = SearchPoint( mapWinID, x, y )
   nHits = CLng(objGeoSelect.Eval("SearchPoint(" & iMapInfoWinID & "," & fXpt &
   "," & fYpt & ")"))

   Set rsGeoREFdata = dbAcropolis.OpenRecordset(Tbl_ValidateUnit, dbOpenTable,
   dbReadOnly)
   rsGeoREFdata.Index = "FACID"

   With frmAriesMain
      !txtPropFacID.Clear
      !trvGISData.Nodes.Clear
      iPropFacIDctr = 0

      If Not (nHits = 0) Then    'there were hits: scan using "SearchInfo"
         ReDim tMapHits(nHits)
```

```
         For i = 1 To nHits
            tMapHits(i).MapTable = objGeoSelect.Eval("SearchInfo(" & i & "," &
   SEARCH_INFO_TABLE & ")")
            iRecNum = CLng(objGeoSelect.Eval("SearchInfo(" & i & "," &
   SEARCH_INFO_ROW & ")"))

            objGeoSelect.Do "Fetch Rec " & iRecNum & " from " &
   tMapHits(i).MapTable
            tMapHits(i).GeoFacID = objGeoSelect.Eval(tMapHits(i).MapTable &
   ".col1")

            !txtPropFacID.AddItem tMapHits(i).GeoFacID
         Next i

         'Get & Display Proposed Facility Data
         tFacilityInfo.FacID = !txtPropFacID.List(0)
         Call GetPropFacilityInfo(tFacilityInfo, rsGeoREFdata)
         Call DisplayPropFacilityInfo(tFacilityInfo)
         Call DisplayTreeViewInfo(tMapHits(), nHits)

         If (nHits = 1) Then
            !spinFacIDbtn.Visible = False
         Else
            !spinFacIDbtn.Visible = True
         End If
      Else     'No Facility Selected - Clear all Proposed Facility fields
         !cboUIC.Clear
         !lstbxUnitName.Clear
         !lstbxUnitName.AddItem "** NO FACILITY SELECTED **"
         !txtCity = vbNullString
         !txtState = vbNullString
         !txtZipCode = vbNullString
      End If
   End With
End Sub

'*************************************************************************
     SetStatusText
'       This is a standard method that MapInfo looks for in its Callback
'       object. If MapInfo finds this method in the Callback object, it
'       will get called automatically every time the text in MapInfo's
'       status bar text changes. The string passed in is a tab-delimited
'       string containing each of the status bar "Fields", i.e. - the
'       first field is the zoom, scale, or cursor location, the second field
'       is the editable layer, etc.
'*************************************************************************

Public Sub SetStatusText(ByVal cStatusText As String)
   cProcName = "SET_STATUS_TEXT"

   Dim cNewText As String

   cNewText = GetField(cStatusText, vbTab, 1)        '* get first field in tab-
   delimited string

   If Not (cNewText = "") Then
      frmAriesMain!AriesStatusBar.Panels(4).Text = cNewText
   End If

End Sub
```

THIS PAGE LEFT INTENTIONALLY BLANK

Module 7. MAPBASIC PUBLIC DECLARATIONS

Purpose: A library module that initializes all public variables and constants for communicating with the integrated mapping component.

Source Type: Visual Basic for Applications

Source File: libMAPBASIC.BAS

Code Listing:

```
Attribute VB_Name = "libMAPBASIC"

'=============================================================================
' MapInfo version 4.0 - System defines
'-----------------------------------------------------------------------------
' This file contains defines useful when programming in the MapBasic
' language.  There are three versions of this file:
'     MAPBASIC.DEF  - MapBasic syntax
'     MAPBASIC.BAS  - Visual Basic syntax
'     MAPBASIC.H    - C/C++ syntax
'-----------------------------------------------------------------------------
' The defines in this file are organized into the following sections:
'     General Purpose defines:
'             Macros, Logical constants, Angle conversion, Colors
'     ButtonPadInfo() defines
'     ColumnInfo() defines
'     CommandInfo() defines
'     FileAttr() defines
'     IntersectNodes() parameters
'     LayerInfo() defines
'     MapperInfo() defines
'     MenuItemInfoByID() and MenuItemInfoByHandler() defines
'     ObjectGeography() defines
'     ObjectInfo() defines
'     SearchInfo() defines
'     SelectionInfo() defines
'     Server statement and function defines
'     StringCompare() return values
'     StyleAttr() defines
'     SystemInfo() defines
'     TableInfo() defines
'     WindowInfo() defines
'     Abbreviated list of error codes
'     Backward Compatibility defines
'=============================================================================
' This file is converted into MAPBASIC.H by doing the following:
'   - concatenate MAPBASIC.DEF and MENU.DEF into MAPBASIC.H
'   - search & replace "'" at begining of a line with "//"
'   - search & replace "Define" at begining of a line with "#define"
'   - delete the following sections:
'         * General Purpose defines: Macros, Logical Constants, Angle
   Conversions
'         * Abbreviated list of error codes
'         * Backward Compatibility defines
'         * Menu constants whose names have changed
'         * Obsolete menu items
'=============================================================================
' This file is converted into MAPBASIC.BAS by doing the following:
'   - concatenate MAPBASIC.DEF and MENU.DEF into MAPBASIC.BAS
'   - search & replace "Define <name>" with "Global Const <name> ="
'     e.g. "<Define {[!-z]+} +{[!-z]}" with "Global Const \0 = \1" using Brief
```

```
'   - delete the following sections:
'         * General Purpose defines: Macros, Logical Constants, Angle
    Conversions
'         * Abbreviated list of error codes
'         * Backward Compatibility defines
'         * Menu constants whose names have changed
'         * Obsolete menu items
'=============================================================================

'=============================================================================
' General Purpose defines
'=============================================================================
'-----------------------------------------------------------------------------
' Colors
'-----------------------------------------------------------------------------
Global Const BLACK = 0
Global Const WHITE = 16777215
Global Const RED = 16711680
Global Const GREEN = 65280
Global Const BLUE = 255
Global Const CYAN = 65535
Global Const MAGENTA = 16711935
Global Const YELLOW = 16776960

'=============================================================================
' ButtonPadInfo() defines
'=============================================================================
Global Const BTNPAD_INFO_FLOATING = 1
Global Const BTNPAD_INFO_WIDTH = 2
Global Const BTNPAD_INFO_NBTNS = 3
Global Const BTNPAD_INFO_X = 4
Global Const BTNPAD_INFO_Y = 5
Global Const BTNPAD_INFO_WINID = 6

'=============================================================================
' ColumnInfo() defines
'=============================================================================
Global Const COL_INFO_NAME = 1
Global Const COL_INFO_NUM = 2
Global Const COL_INFO_TYPE = 3
Global Const COL_INFO_WIDTH = 4
Global Const COL_INFO_DECPLACES = 5
Global Const COL_INFO_INDEXED = 6
Global Const COL_INFO_EDITABLE = 7

'-----------------------------------------------------------------------------
' Column type defines, returned by ColumnInfo(<col_ref>, COL_INFO_TYPE)
'-----------------------------------------------------------------------------
Global Const COL_TYPE_CHAR = 1
Global Const COL_TYPE_DECIMAL = 2
Global Const COL_TYPE_INTEGER = 3
Global Const COL_TYPE_SMALLINT = 4
Global Const COL_TYPE_DATE = 5
Global Const COL_TYPE_LOGICAL = 6
Global Const COL_TYPE_GRAPHIC = 7
Global Const COL_TYPE_FLOAT = 8

'=============================================================================
' CommandInfo() defines
'=============================================================================
Global Const CMD_INFO_X = 1
Global Const CMD_INFO_Y = 2
Global Const CMD_INFO_SHIFT = 3
Global Const CMD_INFO_CTRL = 4
Global Const CMD_INFO_X2 = 5
Global Const CMD_INFO_Y2 = 6
Global Const CMD_INFO_TOOLBTN = 7
Global Const CMD_INFO_MENUITEM = 8
Global Const CMD_INFO_WIN = 1
Global Const CMD_INFO_SELTYPE = 1
```

```
Global Const CMD_INFO_ROWID = 2
Global Const CMD_INFO_INTERRUPT = 3
Global Const CMD_INFO_STATUS = 1
Global Const CMD_INFO_MSG = 1000
Global Const CMD_INFO_DLG_OK = 1
Global Const CMD_INFO_DLG_DBL = 1
Global Const CMD_INFO_FIND_RC = 3
Global Const CMD_INFO_FIND_ROWID = 4
Global Const CMD_INFO_XCMD = 1
Global Const CMD_INFO_CUSTOM_OBJ = 1
Global Const CMD_INFO_TASK_SWITCH = 1

'-------------------------------------------------------------------------
' Task Switch, returned by CommandInfo(CMD_INFO_TASK_SWITCH)
'-------------------------------------------------------------------------
Global Const SWITCHING_OUT_OF_MAPINFO = 0
Global Const SWITCHING_INTO_MAPINFO = 1

'=========================================================================
' FileAttr() defines
'=========================================================================
Global Const FILE_ATTR_MODE = 1
Global Const FILE_ATTR_FILESIZE = 2

'-------------------------------------------------------------------------
' File Access modes, returned by FileAttr(<file_id>, FILE_ATTR_MODE)
'-------------------------------------------------------------------------
Global Const MODE_INPUT = 0
Global Const MODE_OUTPUT = 1
Global Const MODE_APPEND = 2
Global Const MODE_RANDOM = 3
Global Const MODE_BINARY = 4

'=========================================================================
' IntersectNodes(obj1, obj2, mode) parameters
'=========================================================================
Global Const INCL_CROSSINGS = 1
Global Const INCL_COMMON = 6
Global Const INCL_ALL = 7

'=========================================================================
' LayerInfo() defines
'=========================================================================
Global Const LAYER_INFO_NAME = 1
Global Const LAYER_INFO_EDITABLE = 2
Global Const LAYER_INFO_SELECTABLE = 3
Global Const LAYER_INFO_ZOOM_LAYERED = 4
Global Const LAYER_INFO_ZOOM_MIN = 5
Global Const LAYER_INFO_ZOOM_MAX = 6
Global Const LAYER_INFO_COSMETIC = 7
Global Const LAYER_INFO_PATH = 8
Global Const LAYER_INFO_DISPLAY = 9
Global Const LAYER_INFO_OVR_LINE = 10
Global Const LAYER_INFO_OVR_PEN = 11
Global Const LAYER_INFO_OVR_BRUSH = 12
Global Const LAYER_INFO_OVR_SYMBOL = 13
Global Const LAYER_INFO_OVR_FONT = 14
Global Const LAYER_INFO_LBL_EXPR = 15
Global Const LAYER_INFO_LBL_LT = 16
Global Const LAYER_INFO_LBL_CURFONT = 17
Global Const LAYER_INFO_LBL_FONT = 18
Global Const LAYER_INFO_LBL_PARALLEL = 19
Global Const LAYER_INFO_LBL_POS = 20
Global Const LAYER_INFO_ARROWS = 21
Global Const LAYER_INFO_NODES = 22
Global Const LAYER_INFO_CENTROIDS = 23
Global Const LAYER_INFO_TYPE = 24
Global Const LAYER_INFO_LBL_VISIBILITY = 25
Global Const LAYER_INFO_LBL_ZOOM_MIN = 26
Global Const LAYER_INFO_LBL_ZOOM_MAX = 27
```

```
Global Const LAYER_INFO_LBL_AUTODISPLAY = 28
Global Const LAYER_INFO_LBL_OVERLAP = 29
Global Const LAYER_INFO_LBL_DUPLICATES = 30
Global Const LAYER_INFO_LBL_OFFSET = 31
Global Const LAYER_INFO_LBL_MAX = 32

'-------------------------------------------------------------------------
' Display Modes, returned by LayerInfo() for LAYER_INFO_DISPLAY
'-------------------------------------------------------------------------
Global Const LAYER_INFO_DISPLAY_OFF = 0
Global Const LAYER_INFO_DISPLAY_GRAPHIC = 1
Global Const LAYER_INFO_DISPLAY_GLOBAL = 2
Global Const LAYER_INFO_DISPLAY_VALUE = 3

'-------------------------------------------------------------------------
' Label Linetypes, returned by LayerInfo() for LAYER_INFO_LBL_LT
'-------------------------------------------------------------------------
Global Const LAYER_INFO_LBL_LT_NONE = 0
Global Const LAYER_INFO_LBL_LT_SIMPLE = 1
Global Const LAYER_INFO_LBL_LT_ARROW = 2

'-------------------------------------------------------------------------
' Label Positions, returned by LayerInfo() for LAYER_INFO_LBL_POS
'-------------------------------------------------------------------------
Global Const LAYER_INFO_LBL_POS_CC = 0
Global Const LAYER_INFO_LBL_POS_TL = 1
Global Const LAYER_INFO_LBL_POS_TC = 2
Global Const LAYER_INFO_LBL_POS_TR = 3
Global Const LAYER_INFO_LBL_POS_CL = 4
Global Const LAYER_INFO_LBL_POS_CR = 5
Global Const LAYER_INFO_LBL_POS_BL = 6
Global Const LAYER_INFO_LBL_POS_BC = 7
Global Const LAYER_INFO_LBL_POS_BR = 8

'-------------------------------------------------------------------------
' Layer Types, returned by LayerInfo() for LAYER_INFO_TYPE
'-------------------------------------------------------------------------
Global Const LAYER_INFO_TYPE_NORMAL = 0
Global Const LAYER_INFO_TYPE_COSMETIC = 1
Global Const LAYER_INFO_TYPE_IMAGE = 2
Global Const LAYER_INFO_TYPE_THEMATIC = 3

'-------------------------------------------------------------------------
' Label visibility modes, returned by LayerInfo() for LAYER_INFO_LBL_VISIBILITY
'-------------------------------------------------------------------------
Global Const LAYER_INFO_LBL_VIS_OFF = 1
Global Const LAYER_INFO_LBL_VIS_ZOOM = 2
Global Const LAYER_INFO_LBL_VIS_ON = 3

'=========================================================================
' MapperInfo() defines
'=========================================================================
Global Const MAPPER_INFO_ZOOM = 1
Global Const MAPPER_INFO_SCALE = 2
Global Const MAPPER_INFO_CENTERX = 3
Global Const MAPPER_INFO_CENTERY = 4
Global Const MAPPER_INFO_MINX = 5
Global Const MAPPER_INFO_MINY = 6
Global Const MAPPER_INFO_MAXX = 7
Global Const MAPPER_INFO_MAXY = 8
Global Const MAPPER_INFO_LAYERS = 9
Global Const MAPPER_INFO_EDIT_LAYER = 10
Global Const MAPPER_INFO_XYUNITS = 11
Global Const MAPPER_INFO_DISTUNITS = 12
Global Const MAPPER_INFO_AREAUNITS = 13
Global Const MAPPER_INFO_SCROLLBARS = 14
Global Const MAPPER_INFO_DISPLAY = 15
Global Const MAPPER_INFO_NUM_THEMATIC = 16
Global Const MAPPER_INFO_COORDSYS_CLAUSE = 17
Global Const MAPPER_INFO_COORDSYS_NAME = 18
```

```
'-------------------------------------------------------------------------
' Display Modes, returned by MapperInfo() for MAPPER_INFO_DISPLAY
'-------------------------------------------------------------------------
Global Const MAPPER_INFO_DISPLAY_SCALE = 0
Global Const MAPPER_INFO_DISPLAY_ZOOM = 1
Global Const MAPPER_INFO_DISPLAY_POSITION = 2

'=========================================================================
' MenuItemInfoByID() and MenuItemInfoByHandler() defines
'=========================================================================
Global Const MENUITEM_INFO_ENABLED = 1
Global Const MENUITEM_INFO_CHECKED = 2
Global Const MENUITEM_INFO_CHECKABLE = 3
Global Const MENUITEM_INFO_SHOWHIDEABLE = 4
Global Const MENUITEM_INFO_ACCELERATOR = 5
Global Const MENUITEM_INFO_TEXT = 6
Global Const MENUITEM_INFO_HELPMSG = 7
Global Const MENUITEM_INFO_HANDLER = 8
Global Const MENUITEM_INFO_ID = 9

'=========================================================================
' ObjectGeography() defines
'=========================================================================
Global Const OBJ_GEO_MINX = 1
Global Const OBJ_GEO_LINEBEGX = 1
Global Const OBJ_GEO_POINTX = 1
Global Const OBJ_GEO_MINY = 2
Global Const OBJ_GEO_LINEBEGY = 2
Global Const OBJ_GEO_POINTY = 2
Global Const OBJ_GEO_MAXX = 3
Global Const OBJ_GEO_LINEENDX = 3
Global Const OBJ_GEO_MAXY = 4
Global Const OBJ_GEO_LINEENDY = 4
Global Const OBJ_GEO_ARCBEGANGLE = 5
Global Const OBJ_GEO_TEXTLINEX = 5
Global Const OBJ_GEO_ROUNDRADIUS = 5
Global Const OBJ_GEO_ARCENDANGLE = 6
Global Const OBJ_GEO_TEXTLINEY = 6
Global Const OBJ_GEO_TEXTANGLE = 7

'=========================================================================
' ObjectInfo() defines
'=========================================================================
Global Const OBJ_INFO_TYPE = 1
Global Const OBJ_INFO_PEN = 2
Global Const OBJ_INFO_SYMBOL = 2
Global Const OBJ_INFO_TEXTFONT = 2
Global Const OBJ_INFO_BRUSH = 3
Global Const OBJ_INFO_NPNTS = 20
Global Const OBJ_INFO_TEXTSTRING = 3
Global Const OBJ_INFO_SMOOTH = 4
Global Const OBJ_INFO_FRAMEWIN = 4
Global Const OBJ_INFO_NPOLYGONS = 21
Global Const OBJ_INFO_TEXTSPACING = 4
Global Const OBJ_INFO_TEXTJUSTIFY = 5
Global Const OBJ_INFO_FRAMETITLE = 6
Global Const OBJ_INFO_TEXTARROW = 6

'-------------------------------------------------------------------------
' Object types, returned by ObjectInfo(<obj>, OBJ_INFO_TYPE)
'-------------------------------------------------------------------------
Global Const OBJ_TYPE_ARC = 1
Global Const OBJ_TYPE_ELLIPSE = 2
Global Const OBJ_TYPE_LINE = 3
Global Const OBJ_TYPE_PLINE = 4
Global Const OBJ_TYPE_POINT = 5
Global Const OBJ_TYPE_FRAME = 6
Global Const OBJ_TYPE_REGION = 7
Global Const OBJ_TYPE_RECT = 8
```

```
Global Const OBJ_TYPE_ROUNDRECT = 9
Global Const OBJ_TYPE_TEXT = 10

'=============================================================================
' SearchInfo() defines
'=============================================================================
Global Const SEARCH_INFO_TABLE = 1
Global Const SEARCH_INFO_ROW = 2

'=============================================================================
' SelectionInfo() defines
'=============================================================================
Global Const SEL_INFO_TABLENAME = 1
Global Const SEL_INFO_SELNAME = 2
Global Const SEL_INFO_NROWS = 3

'=============================================================================
' Server statement and function defines
'=============================================================================
'-----------------------------------------------------------------------------
' Return Codes
'-----------------------------------------------------------------------------
Global Const SRV_SUCCESS = 0
Global Const SRV_SUCCESS_WITH_INFO = 1
Global Const SRV_ERROR = -1
Global Const SRV_INVALID_HANDLE = -2
Global Const SRV_NEED_DATA = 99
Global Const SRV_NO_MORE_DATA = 100

'-----------------------------------------------------------------------------
' Special values for the status associated with a fetched value
'-----------------------------------------------------------------------------
Global Const SRV_NULL_DATA = -1
Global Const SRV_TRUNCATED_DATA = -2

'-----------------------------------------------------------------------------
' Server_ColumnInfo() Attr defines
'-----------------------------------------------------------------------------
Global Const SRV_COL_INFO_NAME = 1
Global Const SRV_COL_INFO_TYPE = 2
Global Const SRV_COL_INFO_WIDTH = 3
Global Const SRV_COL_INFO_PRECISION = 4
Global Const SRV_COL_INFO_SCALE = 5
Global Const SRV_COL_INFO_VALUE = 6
Global Const SRV_COL_INFO_STATUS = 7

'-----------------------------------------------------------------------------
' Column types, returned by Server_ColumnInfo(<stmt>,<colno>,SRV_COL_INFO_TYPE)
'-----------------------------------------------------------------------------
Global Const SRV_COL_TYPE_NONE = 0
Global Const SRV_COL_TYPE_CHAR = 1
Global Const SRV_COL_TYPE_DECIMAL = 2
Global Const SRV_COL_TYPE_INTEGER = 3
Global Const SRV_COL_TYPE_SMALLINT = 4
Global Const SRV_COL_TYPE_DATE = 5
Global Const SRV_COL_TYPE_LOGICAL = 6
Global Const SRV_COL_TYPE_FLOAT = 8
Global Const SRV_COL_TYPE_FIXED_LEN_STRING = 16
Global Const SRV_COL_TYPE_BIN_STRING = 17

'-----------------------------------------------------------------------------
' Server_DriverInfo() Attr defines
'-----------------------------------------------------------------------------
Global Const SRV_DRV_INFO_NAME = 1
Global Const SRV_DRV_INFO_NAME_LIST = 2
Global Const SRV_DRV_DATA_SOURCE = 3

'-----------------------------------------------------------------------------
' Fetch Directions used by Server_Fetch()
'-----------------------------------------------------------------------------
```

```
Global Const SRV_FETCH_NEXT = -1
Global Const SRV_FETCH_PREV = -2
Global Const SRV_FETCH_FIRST = -3
Global Const SRV_FETCH_LAST = -4

'================================================================================
' StringCompare(<str_1>, <str_2>) return values
'================================================================================
Global Const STR_LT = -1
Global Const STR_GT = 1
Global Const STR_EQ = 0

'================================================================================
' StyleAttr() defines
'================================================================================
Global Const PEN_WIDTH = 1
Global Const PEN_PATTERN = 2
Global Const PEN_COLOR = 4
Global Const BRUSH_PATTERN = 1
Global Const BRUSH_FORECOLOR = 2
Global Const BRUSH_BACKCOLOR = 3
Global Const FONT_NAME = 1
Global Const FONT_STYLE = 2
Global Const FONT_POINTSIZE = 3
Global Const FONT_FORECOLOR = 4
Global Const FONT_BACKCOLOR = 5
Global Const SYMBOL_CODE = 1
Global Const SYMBOL_COLOR = 2
Global Const SYMBOL_POINTSIZE = 3
Global Const SYMBOL_ANGLE = 4
Global Const SYMBOL_FONT_NAME = 5
Global Const SYMBOL_FONT_STYLE = 6
Global Const SYMBOL_KIND = 7
Global Const SYMBOL_CUSTOM_NAME = 8
Global Const SYMBOL_CUSTOM_STYLE = 9

'--------------------------------------------------------------------------------
' Symbol kinds returned by StyleAttr(<symbol>, SYMBOL_KIND)
'--------------------------------------------------------------------------------
Global Const SYMBOL_KIND_VECTOR = 1
Global Const SYMBOL_KIND_FONT = 2
Global Const SYMBOL_KIND_CUSTOM = 3

'================================================================================
' SystemInfo() defines
'================================================================================
Global Const SYS_INFO_PLATFORM = 1
Global Const SYS_INFO_APPVERSION = 2
Global Const SYS_INFO_MIVERSION = 3
Global Const SYS_INFO_RUNTIME = 4
Global Const SYS_INFO_CHARSET = 5
Global Const SYS_INFO_COPYPROTECTED = 6
Global Const SYS_INFO_APPLICATIONWND = 7
Global Const SYS_INFO_DDESTATUS = 8
Global Const SYS_INFO_MAPINFOWND = 9
Global Const SYS_INFO_NUMBER_FORMAT = 10
Global Const SYS_INFO_DATE_FORMAT = 11
Global Const SYS_INFO_DIG_INSTALLED = 12
Global Const SYS_INFO_DIG_MODE = 13
Global Const SYS_INFO_MIPLATFORM = 14
Global Const SYS_INFO_MDICLIENTWND = 15

'--------------------------------------------------------------------------------
' Platform, returned by SystemInfo(SYS_INFO_PLATFORM)
'--------------------------------------------------------------------------------
Global Const PLATFORM_SPECIAL = 0
Global Const PLATFORM_WIN = 1
Global Const PLATFORM_MAC = 2
Global Const PLATFORM_MOTIF = 3
Global Const PLATFORM_X11 = 4
```

```
Global Const PLATFORM_X0L = 5

'------------------------------------------------------------------------------
' Version, returned by SystemInfo(SYS_INFO_MIPLATFORM)
'------------------------------------------------------------------------------
Global Const MIPLATFORM_SPECIAL = 0
Global Const MIPLATFORM_WIN16 = 1
Global Const MIPLATFORM_WIN32 = 2
Global Const MIPLATFORM_POWERMAC = 3
Global Const MIPLATFORM_MAC68K = 4
Global Const MIPLATFORM_HP = 5
Global Const MIPLATFORM_SUN = 6

'==============================================================================
' TableInfo() defines
'==============================================================================
Global Const TAB_INFO_NAME = 1
Global Const TAB_INFO_NUM = 2
Global Const TAB_INFO_TYPE = 3
Global Const TAB_INFO_NCOLS = 4
Global Const TAB_INFO_MAPPABLE = 5
Global Const TAB_INFO_READONLY = 6
Global Const TAB_INFO_TEMP = 7
Global Const TAB_INFO_NROWS = 8
Global Const TAB_INFO_EDITED = 9
Global Const TAB_INFO_FASTEDIT = 10
Global Const TAB_INFO_UNDO = 11
Global Const TAB_INFO_MAPPABLE_TABLE = 12
Global Const TAB_INFO_USERMAP = 13
Global Const TAB_INFO_USERBROWSE = 14
Global Const TAB_INFO_USERCLOSE = 15
Global Const TAB_INFO_USEREDITABLE = 16
Global Const TAB_INFO_USERREMOVEMAP = 17
Global Const TAB_INFO_USERDISPLAYMAP = 18
Global Const TAB_INFO_TABFILE = 19
Global Const TAB_INFO_MINX = 20
Global Const TAB_INFO_MINY = 21
Global Const TAB_INFO_MAXX = 22
Global Const TAB_INFO_MAXY = 23
Global Const TAB_INFO_SEAMLESS = 24
Global Const TAB_INFO_COORDSYS_MINX = 25
Global Const TAB_INFO_COORDSYS_MINY = 26
Global Const TAB_INFO_COORDSYS_MAXX = 27
Global Const TAB_INFO_COORDSYS_MAXY = 28
Global Const TAB_INFO_COORDSYS_CLAUSE = 29
Global Const TAB_INFO_COORDSYS_NAME = 30
Global Const TAB_INFO_NREFS = 31

'------------------------------------------------------------------------------
' Table type defines, returned by TableInfo(<tab_ref>, TAB_INFO_TYPE)
'------------------------------------------------------------------------------
Global Const TAB_TYPE_BASE = 1
Global Const TAB_TYPE_RESULT = 2
Global Const TAB_TYPE_VIEW = 3
Global Const TAB_TYPE_IMAGE = 4
Global Const TAB_TYPE_LINKED = 5

'==============================================================================
' WindowInfo() defines
'==============================================================================
Global Const WIN_INFO_NAME = 1
Global Const WIN_INFO_TYPE = 3
Global Const WIN_INFO_WIDTH = 4
Global Const WIN_INFO_HEIGHT = 5
Global Const WIN_INFO_X = 6
Global Const WIN_INFO_Y = 7
Global Const WIN_INFO_TOPMOST = 8
Global Const WIN_INFO_STATE = 9
Global Const WIN_INFO_TABLE = 10
Global Const WIN_INFO_LEGENDS_MAP = 10
```

```
Global Const WIN_INFO_OPEN = 11
Global Const WIN_INFO_WND = 12
Global Const WIN_INFO_WINDOWID = 13
Global Const WIN_INFO_WORKSPACE = 14
Global Const WIN_INFO_CLONEWINDOW = 15
Global Const WIN_INFO_SYSMENUCLOSE = 16
Global Const WIN_INFO_AUTOSCROLL = 17

'-------------------------------------------------------------------------
' Window types, returned by WindowInfo(<win_id>, WIN_INFO_TYPE)
'-------------------------------------------------------------------------
Global Const WIN_MAPPER = 1
Global Const WIN_BROWSER = 2
Global Const WIN_LAYOUT = 3
Global Const WIN_GRAPH = 4
Global Const WIN_BUTTONPAD = 19
Global Const WIN_HELP = 1001
Global Const WIN_MAPBASIC = 1002
Global Const WIN_MESSAGE = 1003
Global Const WIN_RULER = 1007
Global Const WIN_INFO = 1008
Global Const WIN_LEGEND = 1009
Global Const WIN_STATISTICS = 1010
Global Const WIN_MAPINFO = 1011
'-------------------------------------------------------------------------
' Version 2 window types no longer used in version 3 or version 4
'-------------------------------------------------------------------------
Global Const WIN_TOOLPICKER = 1004
Global Const WIN_PENPICKER = 1005
Global Const WIN_SYMBOLPICKER = 1006

'-------------------------------------------------------------------------
' Window states, returned by WindowInfo(<win_id>, WIN_INFO_STATE)
'-------------------------------------------------------------------------
Global Const WIN_STATE_NORMAL = 0
Global Const WIN_STATE_MINIMIZED = 1
Global Const WIN_STATE_MAXIMIZED = 2

'=========================================================================
' Set Next Document Style defines
'=========================================================================
Global Const WIN_STYLE_STANDARD = 0
Global Const WIN_STYLE_CHILD = 1
Global Const WIN_STYLE_POPUP_FULLCAPTION = 2
Global Const WIN_STYLE_POPUP = 3

'=========================================================================
' end of MAPBASIC.DEF
'=========================================================================

'=========================================================================
' MapInfo version 4.0 - Menu Item Definitions
'-------------------------------------------------------------------------
' This file contains defines useful when programming in the MapBasic
' language.  The definitions in this file describe the standard MapInfo
' functionality available via the "Run Menu Command" MapBasic statement.
'-------------------------------------------------------------------------
' The defines in this file are organized to match the sequence of
' declarations in the MAPINFOW.MNU file, which in turn reflects the
' organization of the MapInfo menus and buttonpads.
'=========================================================================

'-------------------------------------------------------------------------
' File & Send Mail menus
'-------------------------------------------------------------------------
Global Const M_FILE_NEW = 101
Global Const M_FILE_OPEN = 102
Global Const M_FILE_OPEN_ODBC = 116
Global Const M_FILE_ADD_WORKSPACE = 108
Global Const M_FILE_CLOSE = 103
```

```
Global Const M_FILE_CLOSE_ALL = 104
Global Const M_FILE_SAVE = 105
Global Const M_FILE_SAVE_COPY_AS = 106
Global Const M_FILE_SAVE_WORKSPACE = 109
Global Const M_FILE_SAVE_WINDOW_AS = 609
Global Const M_FILE_REVERT = 107
Global Const M_FILE_RUN = 110
Global Const M_FILE_PAGE_SETUP = 111
Global Const M_FILE_PRINT = 112
Global Const M_FILE_EXIT = 113

Global Const M_SENDMAIL_CURRENTWINDOW = 114
Global Const M_SENDMAIL_WORKSPACE = 115

'-------------------------------------------------------------------------------
' Edit menu
'-------------------------------------------------------------------------------
Global Const M_EDIT_UNDO = 201
Global Const M_EDIT_CUT = 202
Global Const M_EDIT_COPY = 203
Global Const M_EDIT_PASTE = 204
Global Const M_EDIT_CLEAR = 205
Global Const M_EDIT_CLEAROBJ = 206
Global Const M_EDIT_RESHAPE = 1601
Global Const M_EDIT_NEW_ROW = 702
Global Const M_EDIT_GETINFO = 207

'-------------------------------------------------------------------------------
' Objects menu
'-------------------------------------------------------------------------------
Global Const M_OBJECTS_SET_TARGET = 1610
Global Const M_OBJECTS_CLEAR_TARGET = 1611
Global Const M_OBJECTS_COMBINE = 1605
Global Const M_OBJECTS_SPLIT = 1612
Global Const M_OBJECTS_ERASE = 1613
Global Const M_OBJECTS_ERASE_OUT = 1614
Global Const M_OBJECTS_OVERLAY = 1615
Global Const M_OBJECTS_BUFFER = 1606
Global Const M_OBJECTS_SMOOTH = 1602
Global Const M_OBJECTS_UNSMOOTH = 1603
Global Const M_OBJECTS_CVT_PGON = 1607
Global Const M_OBJECTS_CVT_PLINE = 1604

'-------------------------------------------------------------------------------
' Query menu
'-------------------------------------------------------------------------------
Global Const M_ANALYZE_SELECT = 301
Global Const M_ANALYZE_SQLQUERY = 302
Global Const M_ANALYZE_SELECTALL = 303
Global Const M_ANALYZE_UNSELECT = 304
Global Const M_ANALYZE_FIND = 305
Global Const M_ANALYZE_FIND_SELECTION = 306
Global Const M_ANALYZE_CALC_STATISTICS = 309

'-------------------------------------------------------------------------------
' Table, Maintenance, and Raster menus
'-------------------------------------------------------------------------------
Global Const M_TABLE_UPDATE_COLUMN = 405
Global Const M_TABLE_APPEND = 411
Global Const M_TABLE_GEOCODE = 407
Global Const M_TABLE_CREATE_POINTS = 408
Global Const M_TABLE_MERGE_USING_COLUMN = 406
Global Const M_TABLE_IMPORT = 401
Global Const M_TABLE_EXPORT = 402

Global Const M_TABLE_MODIFY_STRUCTURE = 404
Global Const M_TABLE_DELETE = 409
Global Const M_TABLE_RENAME = 410
Global Const M_TABLE_PACK = 403
Global Const M_TABLE_MAKEMAPPABLE = 415
```

```
Global Const M_TABLE_UNLINK = 416
Global Const M_TABLE_REFRESH = 417

Global Const M_TABLE_RASTER_STYLE = 414
Global Const M_TABLE_RASTER_REG = 413
Global Const M_TOOLS_RASTER_REG = 1730

'-------------------------------------------------------------------------
' Options menu
'-------------------------------------------------------------------------
Global Const M_FORMAT_PICK_LINE = 501
Global Const M_FORMAT_PICK_FILL = 502
Global Const M_FORMAT_PICK_SYMBOL = 503
Global Const M_FORMAT_PICK_FONT = 504
Global Const M_WINDOW_BUTTONPAD = 605
Global Const M_WINDOW_LEGEND = 606
Global Const M_WINDOW_STATISTICS = 607
Global Const M_WINDOW_MAPBASIC = 608
Global Const M_WINDOW_STATUSBAR = 616
Global Const M_FORMAT_CUSTOM_COLORS = 617
Global Const M_EDIT_PREFERENCES = 208
Global Const M_EDIT_PREFERENCES_SYSTEM = 210
Global Const M_EDIT_PREFERENCES_FILE = 211
Global Const M_EDIT_PREFERENCES_MAP = 212
Global Const M_EDIT_PREFERENCES_COUNTRY = 213
Global Const M_EDIT_PREFERENCES_PATH = 214

'-------------------------------------------------------------------------
' Window menu
'-------------------------------------------------------------------------
Global Const M_WINDOW_BROWSE = 601
Global Const M_WINDOW_MAP = 602
Global Const M_WINDOW_GRAPH = 603
Global Const M_WINDOW_LAYOUT = 604
Global Const M_WINDOW_REDISTRICT = 615
Global Const M_WINDOW_REDRAW = 610
Global Const M_WINDOW_TILE = 611
Global Const M_WINDOW_CASCADE = 612
Global Const M_WINDOW_ARRANGEICONS = 613
Global Const M_WINDOW_MORE = 614
Global Const M_WINDOW_FIRST = 620
' - - -- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
' Note: the 2nd through 80th windows can be accessed as (M_WINDOW_FIRST+i-1)
' - - -- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

'-------------------------------------------------------------------------
' Help menu
'-------------------------------------------------------------------------
Global Const M_HELP_CONTENTS = 1202
Global Const M_HELP_SEARCH = 1203
Global Const M_HELP_USE_HELP = 1204
Global Const M_HELP_TECHSUPPORT = 1208
Global Const M_HELP_CONNECT_MIFORUM = 1209
Global Const M_HELP_ABOUT = 1205

Global Const M_HELP_CONTEXTSENSITIVE = 1201
Global Const M_HELP_HELPMODE = 1206

'-------------------------------------------------------------------------
' Browse menu
'-------------------------------------------------------------------------
Global Const M_BROWSE_PICK_FIELDS = 704
Global Const M_BROWSE_OPTIONS = 703

'-------------------------------------------------------------------------
' Map menu
'-------------------------------------------------------------------------
Global Const M_MAP_LAYER_CONTROL = 801
Global Const M_MAP_THEMATIC = 307
Global Const M_MAP_MODIFY_THEMATIC = 308
```

```
Global Const M_MAP_CHANGE_VIEW = 805
Global Const M_MAP_CLONE_MAPPER = 811
Global Const M_MAP_PREVIOUS = 806
Global Const M_MAP_ENTIRE_LAYER = 807
Global Const M_MAP_CLEAR_CUSTOM_LABELS = 814
Global Const M_MAP_SAVE_COSMETIC = 809
Global Const M_MAP_CLEAR_COSMETIC = 810
Global Const M_MAP_SET_CLIP_REGION = 812
Global Const M_MAP_CLIP_REGION_ONOFF = 813
Global Const M_MAP_SETUPDIGITIZER = 803
Global Const M_MAP_OPTIONS = 802

'-------------------------------------------------------------------------------
' Layout menu
'-------------------------------------------------------------------------------
Global Const M_LAYOUT_CHANGE_VIEW = 902
Global Const M_LAYOUT_ACTUAL = 903
Global Const M_LAYOUT_ENTIRE = 904
Global Const M_LAYOUT_PREVIOUS = 905
Global Const M_LAYOUT_BRING2FRONT = 906
Global Const M_LAYOUT_SEND2BACK = 907
Global Const M_LAYOUT_ALIGN = 908
Global Const M_LAYOUT_DROPSHADOWS = 909
Global Const M_LAYOUT_DISPLAYOPTIONS = 901

'-------------------------------------------------------------------------------
' Graph menu
'-------------------------------------------------------------------------------
Global Const M_GRAPH_TYPE = 1001
Global Const M_GRAPH_LABEL_AXIS = 1002
Global Const M_GRAPH_VALUE_AXIS = 1003
Global Const M_GRAPH_SERIES = 1004

'-------------------------------------------------------------------------------
' MapBasic menu
'-------------------------------------------------------------------------------
Global Const M_MAPBASIC_CLEAR = 1101
Global Const M_MAPBASIC_SAVECONTENTS = 1102

'-------------------------------------------------------------------------------
' Redistrict menu
'-------------------------------------------------------------------------------
Global Const M_REDISTRICT_ASSIGN = 705
Global Const M_REDISTRICT_TARGET = 706
Global Const M_REDISTRICT_ADD = 707
Global Const M_REDISTRICT_DELETE = 708
Global Const M_REDISTRICT_OPTIONS = 709

'-------------------------------------------------------------------------------
' Main Buttonpad
'-------------------------------------------------------------------------------
Global Const M_TOOLS_SELECTOR = 1701
Global Const M_TOOLS_SEARCH_RECT = 1722
Global Const M_TOOLS_SEARCH_RADIUS = 1703
Global Const M_TOOLS_SEARCH_BOUNDARY = 1704
Global Const M_TOOLS_EXPAND = 1705
Global Const M_TOOLS_SHRINK = 1706
Global Const M_TOOLS_RECENTER = 1702
Global Const M_TOOLS_PNT_QUERY = 1707
Global Const M_TOOLS_LABELER = 1708
Global Const M_TOOLS_DRAGWINDOW = 1734
Global Const M_TOOLS_RULER = 1710

'-------------------------------------------------------------------------------
' Drawing Buttonpad
'-------------------------------------------------------------------------------
Global Const M_TOOLS_POINT = 1711
Global Const M_TOOLS_LINE = 1712
Global Const M_TOOLS_POLYLINE = 1713
Global Const M_TOOLS_ARC = 1716
```

```
Global Const M_TOOLS_POLYGON = 1714
Global Const M_TOOLS_ELLIPSE = 1715
Global Const M_TOOLS_RECTANGLE = 1717
Global Const M_TOOLS_ROUNDEDRECT = 1718
Global Const M_TOOLS_TEXT = 1709
Global Const M_TOOLS_FRAME = 1719
Global Const M_TOOLS_ADD_NODE = 1723

'---------------------------------------------------------------------------
' Menu and ButtonPad items that do not appear in the standard menus
'---------------------------------------------------------------------------
Global Const M_TOOLS_MAPBASIC = 1720
Global Const M_TOOLS_SEARCH_POLYGON = 1733

'===========================================================================
' end of MENU.DEF
'===========================================================================
```

THIS PAGE LEFT INTENTIONALLY BLANK

Module 8. SPATIAL SELECTION PROGRAM

Purpose: A MapBasic program that spatially selects the desired data within 50 miles of the specified location and locates the closest Army Reserve support facilities.

Source Type: MapBasic Spatial Application Development Language

Source File: ARIESARCH.MBA

Code Listing:

```
'*****************************************************************************
'
'      ARIES SDSS SPATIAL SELECTION PROGRAM -- PHASE One
'
'      EXTRACTS SELECTED DATA FROM AROUND PROPOSED FACILITY SITE
'
'*****************************************************************************

Include "c:\mapinfo\mapbasic\mapbasic.def"
Include "c:\mapinfo\mapbasic\menu.def"
Include "c:\mapinfo\mapbasic\icons.def"

Declare Sub Main

Global        AMSA_ID,AMSA_Dist,
       ECS_ID, ECS_Dist,
       RZA_ID, RZA_Dist,
       fProposedFacility,fAreaDistance, fAreaFacID, fAreaZipCode,
       fAreaG18,fAreaIRR,fAreaQMA,TargetZip, cPropFacID As String

Sub MAIN
  Dim FacIDLat, FacIDLong                                  As Float
  Dim objAreaBuffer,objDistanceBuffer, objG18Buffer  As Object
  Dim iRowNum                                              As Integer
  Dim cFilePath, AuthStrength                                    As String

'====================================================================
    cFilePath = ApplicationDirectory$()
    Set Map Center (-96,37) Zoom 3500 Units "mi"

    Create Index On US_Zips(Zip_Code)
    Create Index on AMSA(Fac_Zip)
    Create Index on ECS(Fac_Zip)
    Create Index on RZA(Zip)

    Create Index on GEOREF(Fac_ID)
    Create Index on G18CWE(Zip)
    Create Index on IRR(Zipc)
    Create Index on QMA(Zip)
    Create Index on Non_Clos(Zip)

    fAreaDistance = cFilePath & "AriesData\AreaDist.DBF"
    fAreaFacID = cFilePath & "AriesData\AreaFac.DBF"
    fAreaZipCode = cFilePath & "AriesData\AreaZip.DBF"
    fProposedFacility = cFilePath & "AriesData\PropFac.DBF"

    fAreaG18 = cFilePath & "AriesData\AreaG18.DBF"
    fAreaIRR = cFilePath & "AriesData\AreaIRR.DBF"
    fAreaQMA = cFilePath & "AriesData\AreaQMA.DBF"

    Open Table fAreaDistance
```

```
    Delete From AreaDist

    Open Table fAreaFacID
    Delete From AreaFac

    Open Table fAreaZipCode
    Delete From AreaZip

    Open Table fProposedFacility
      cPropFacID = PropFac.Fac_ID
    Pack Table PropFac Data
    Close Table PropFac

    Open Table fAreaG18
    Delete From AreaG18

    Open Table fAreaIRR
    Delete From AreaIRR

    Open Table fAreaQMA
    Delete From AreaQMA


'=========================================================================
'      (1)  DETERMINE LATITUDE & LONGITUDE OF TARGET ZIP CODE
'      (2)  CENTER MAP AT TARGET ZIP
'      (3)  CREATE 50MI RADIUS CIRCLE AROUND TARGET ZIP CODE.
'=========================================================================

   Find Using GEOREF(Fac_ID)
   Find cPropFacID

   FacIDLat = CommandInfo(CMD_INFO_X)
   FacIDLong = CommandInfo(CMD_INFO_Y)

   Set CoordSys Earth
   'Create Point(FacIDLat, FacIDLong) Symbol (59, RED, 24)

   Insert Into Sites (obj) Values (CreatePoint(FacIDLat, FacIDLong))
   Update Sites Set Fac_ID = cPropFacID Where RowID = 1

   Set Map Center (FacIDLat,FacIDLong) Zoom 500 Units "mi"
   Set Distance Units "mi"

   objAreaBuffer = CreateCircle(FacIDLat,FacIDLong,50)
   objG18Buffer = CreateCircle(FacIDLat,FacIDLong,50)
   objDistanceBuffer = CreateCircle(FacIDLat,FacIDLong,300)

'=========================================================================
'      BUILD A TABLE OF ZIPCODES W/IN 50 MI OF THE PROPOSED
'      FACILITY
'=========================================================================

   Select Zip_Code From US_Zips
   Where Obj Within objAreaBuffer order by Zip_Code into TempZip

   Insert Into AreaZip
     Select * From TempZip

   Commit Table AreaZip
   Pack Table AreaZip Data


'=========================================================================
'      BUILD A TABLE OF FACILITY ID'S W/IN 50 MI OF THE PROPOSED
'      FACILITY
'=========================================================================

   Select Fac_ID From GEOREF
   Where Obj Within objAreaBuffer order by Fac_ID into TempFacID
```

```
   Insert Into AreaFac
     Select * From TempFacID

   Commit Table AreaFac
   Pack Table AreaFac Data


'=========================================================================
'      DETERMINE DISTANCE TO NEAREST RECRUIT STATION
'=========================================================================

   Select * From RZA
   Where Obj Within objDistanceBuffer into TempRZA

   Select Distance(CentroidX(obj),CentroidY(obj),FacIDLat,FacIDLong,"mi")
      "Distance"
   From TempRZA order by Distance into TempRZA_Dist

   RZA_Dist = TempRZA_Dist.Distance


'=========================================================================
'      DETERMINE DISTANCE TO NEAREST AMSA LOCATION
'=========================================================================

   Select * From AMSA
   Where Obj Within objDistanceBuffer into TempAMSA

   Select Distance(CentroidX(obj),CentroidY(obj),FacIDLat,FacIDLong,"mi")
      "Distance"
   From TempAMSA order by Distance into TempAMSA_Dist

   AMSA_Dist = TempAMSA_Dist.Distance


'=========================================================================
'      DETERMINE DISTANCE TO NEAREST EQUIPMENT SITE
'=========================================================================

   Select * From ECS
   Where Obj Within objDistanceBuffer into TempECS

   Select Distance(CentroidX(obj),CentroidY(obj),FacIDLat,FacIDLong,"mi")
      "Distance"
   From TempECS order by Distance into TempECS_Dist

   ECS_Dist = TempECS_Dist.Distance


'=========================================================================
'      DETERMINE AUTHORIZED STRENGTH OF THE NAT GUARD UNITS
'      WITHIN 50 MILES OF TARGET ZIP CODE
'=========================================================================
   Select * From NON_CLOS
   Where Obj Within objAreaBuffer into TempNGUnits

   Select SUM(AUTH)"NO_AUTH_NG"
   From TempNGUnits into Strength

      AuthStrength = Strength.NO_AUTH_NG


'=========================================================================
'      OUTPUT AMSA, ECS, RZA DISTANCES TO AREA DISTANCE FILE
'=========================================================================

   Insert into AreaDist
     Values (AMSA_Dist, ECS_Dist, RZA_Dist, AuthStrength)
```

```
    Commit Table AreaDist
    Pack Table AreaDist Data


'=========================================================================
'      BUILD A TABLE OF G18 Personnel W/IN 50 MI OF THE PROPOSED
'      FACILITY
'=========================================================================
    Select UIC, LEFT$(ZIP,5) "ZIPCODE", PRI "MOS" From G18CWE
    Where Obj Within objG18Buffer and PRI <> ""
    Order by UIC, ZIP into TempG18

    Insert Into AreaG18
      Select * From TempG18

    Commit Table AreaG18
    Pack Table AreaG18 Data


'=========================================================================
'      BUILD A TABLE OF IRR Personnel W/IN 50 MI OF THE PROPOSED
'      FACILITY
'=========================================================================

    Select ZIPC "ZIP", LEFT$(PMOS,3) "MOS" From IRR
    Where Obj Within objAreaBuffer and ZIPC <> "" and PMOS <> ""
    Order by Zipc into TempIRR

    Insert Into AreaIRR
      Select * From TempIRR

    Commit Table AreaIRR
    Pack Table AreaIRR Data


'=========================================================================
'      BUILD A TABLE OF @MA Personnel W/IN 50 MI OF THE PROPOSED
'      FACILITY
'=========================================================================

    Select LEFT$(ZIP,5) "ZIPCODE", MWCAT12, MWCAT3A, MBCAT12, MBCAT3A,
      MHCAT12, MHCAT3A From @MA
    Where Obj Within objAreaBuffer
    Order by Zip into Temp@MA

    Insert Into Area@MA
      Select * From Temp@MA

    Commit Table Area@MA
    Pack Table Area@MA Data


'************************************************************************
'      END MAIN PROGRAM
'************************************************************************
    Delete from Sites
    Commit Table Sites
    Pack Table Sites Data

Exit Sub

end SUB
```

LIST OF REFERENCES.

Aronica, Ronald C., and Rimel, Donald E., Jr., "Wrapper Your Legacy Systems: A Systems Architecture Primer," *Datamation*, Vol. 42, No. 12, June 15, 1996, URL: http://www.datamation.com/PlugIn/issues/1996/june15/ASystemsArchitecturePrimer4.html

Betts, R. K., *Military Readiness: Concepts, Choices, Consequeneces.*, Brookings Institution, 1995.

Booch, Grady, *Object Solutions: Managing the Object-Oriented Project.* Addison-Wesley Publishing Company, Inc., Menlo Park, California, 1996.

Brooks, Fred, "No Silver Bullet: Essence and Accidents of Software Engineering," *IEEE Computer*, Vol. 20, No. 4, April 1987, pp. 74-83.

Chappell, David, *The Next Wave: Component Software Enters The Mainstream*, Chappell & Associates White Paper, April 1997, URL: http://www.rational.com/support/-techpapers/nextwave/nextwave.html

Dahlbom, Bo and Lars Mathiassen, "The Future of Our Profession," *Communications of the ACM*, Vol. 40, No. 6, June 1997, pp. 80-89.

Davis, Margaret and Harold Hawley, "Dialogue–Specified Reuse of Domain Engineering Work Products," *1994 Washington Ada Symposium (WA-daS '94)*, Washington, D.C., 1994.

Department of Defense, Office of the Inspector General, *Evaluation of the Reserve Components Automation System*, Report No. 97-019, *Evaluation Report*, November 1, 1996.

Department of Defense, Office of the Inspector General, *Review of Reserve Forces Management Structure*, December 1993.

Department of Defense, Office of the Under Secretary of Defense for Acquisition, *Report of the Defense Science Board Task Force on Military Software*, September 1987.

Department of the Navy, Naval Research Laboratory, "Software Requirements: A Tutorial," *Naval Research Laboratory Technical Report Abstract*, 1996, URL: ftp://ftp.rome.kaman.com/pub/dacs_reports/softreq

Dill, Robert W., *Data Warehousing and Data Quality for a Spatial Decision Support System*, Master's Thesis Naval Postgraduate School, Monterey, California, September 1997.

Dolan, Kevin, "Prototypes: Tools That Can be Used and Misused," *CrossTalk: The Journal of Defense Software Engineering*, Jan 1994, URL: http://www.stsc.hill.af.mil/ CrossTalk/1994/jan/xt94d01g.html

Dolk, Daniel R., et al., *Combining a Decision Model and GIS for Site Location Problems,* Naval Postgraduate School Working Paper, Code SM/Dk, Monterey, California., 93943, October 1996.

Forte, Gene, "Managing Change for Rapid Development," *IEEE Software*,Vol. 14, No. 3, March/April 1997, pp. 120-122.

Garlan, David, et al., "Architectural Mismatch: Why Reuse is so Hard," *IEEE Software*, Vol. 12, No. 6, November 1995, pp. 17-26.

General Accounting Office, Comptroller General of the United States, *Improving Mission Performance Through Strategic Information Management and Technology: Learning From Leading Organizations*, GAO/AIMD-94-115, *Executive Guide*, May 1994.

Gordon, Scott V., and James M. Bieman, "Rapid Prototyping: Lessons Learned," *IEEE Software*, Vol. 12, No. 1, January 1995, pp. 85-95.

Hamilton, Dennis, "Stop Throwing Hardware at Performance," *Datamation*, Vol. 40, No. 19, October 1, 1994, pp. 43-44.

Hurwitz, Judith I., "Leveraging Chaos: A New Infrastructure to Help You Transition to a New Computing Environment," *DBMS Online*, April 1997, URL: http://www.dbmsmag.com/

Joch, Alan, "Resellers Learn Wall Street Smarts: Financial-Industry Solutions," *BYTE*, Vol. 22, No. 7, July 1997, p. 104D.

Knowles, Jim, "Competing Effectively," *Datamation*, Vol. 43, No. 1, January 1997, URL: http://www.datamation.com/PlugIn/issues/1997/jan/01col40.html

Libicki, Martin C., *Standards: The Rough Road to the Common Byte.*, Center for Advanced Concepts and Technology Institute for National Strategic Studies, National Defense Institute, Washington D.C., May 1995.

Linthicum, David S., "The Good, the RAD, and the Ugly," *DBMS online*, February 1997, URL: http://www.dbmsmag.com/9702d07.html

McCarthy, Jim, *Dynamics of Software Development*, Microsoft Press, Redmond, Washington, 1995.

McConnell, Steve C., *Rapid Development*, Microsoft Press, Redmond, Washington, 1996.

Monroe, Robert T., et al., "Architectural Styles, Design Patterns, and Objects," *IEEE Software*, Vol. 14, No. 1, January 1997, pp. 43-52.

Murphy, Mark A., *An Automated Spatial Decision Support System for the Relocation of Army Reserve Units*, Master's Thesis Naval Postgraduate School, Monterey, California, March 1997.

Paige, Emmett, Jr., "Predictable Software: Order Out of Chaos," *CrossTalk: The Journal of Defense Software Engineering*, June 1994, URL: http://www.stsc.hill.af.mil/crosstalk/1994/jun/xt94d066.html

Sarna, David E., Y., and George J. Febish, "Paradigm Shift: Is There Life after Death?", *Datamation*, Vol. 42, No. 13, July 1996, pp. 27-28.

Scharer, Laura, "The Protyping Alternative," *New Paradigms for Software Development*, Ed. William W. Agresti, IEEE Computer Society Press, 1986. pp. 59-68.

Scott Morton, Michael S., "The State of the Art of Reasearch," *The Information Systems Reasearch Challenge Proceedings,* Ed. F. Warren McFarlan, Harvard Business School Press, Boston, Massachusetts, 1984, p. 25.

Spitzer, Tom, "Component Architectures," *DBMS online*, September 1997, URL: http://www.dbmsmag.com

Tapscott, Don, and Art Caston, *Paradigm Shift The New Promise of Information Technology*. McGraw-Hill Inc., San Francisco, California, 1993.

Taylor, David A. *Object-Oriented Technology: A Manager's Guide*. Addison-Wesley Publishing Company, Menlo Park, California, 1990.

Tkach, Daniel, and Richard Puttick, *Object Technology in Application Development*. 2nd ed. Addison-Wesley Publishing Company, Menlo Park, California, 1996.

Udell, Jon, "ComponentWare," *BYTE*, Vol. 19, No. 5, May 1994, pp. 46-56.

United States Air Force, Software Technology Support Center, *Software Management Guide*, 3rd ed. Ed. Raymond J. Rubey, Hill Air Force Base, April 1992.

Whitten, Jeffrey L., Bentley, Lonnie D., and Barlow, Victor M., *System Analysis and Design Methods* 3rd ed. Richard D. Irwin, Inc., Boston Massachusetts, 1994.

Yourdon, Edward, *Rise and Resurrection of the American Programmer*. Yourdon Press, Upper Saddle River, New Jersey, 1996.

BIBLIOGRAPHY.

Amundsen, Michael, and Curtis Smith. *Teach Yourself Database Programming with Visual Basic 4 in 21 days*, Sams Publishing, Indianapolis, Indiana, 1996.

Appleman, Daniel. *Visual Basic Programmer's Guide to the Win32 API*, Ziff-Davis Press, Emeryville, California, 1996.

Application Developers Training Company, *Fundamentals of Programming in Visual Basic*, 1996.

Application Developers Training Company, *Intermediate Programming in Visual Basic*, 1996.

Application Developers Training Company, *Advanced Programming in Visual Basic*, 1996.

Arnold, Thomas R., and William A. Fuson, "Testing in a Perfect World," *Communications of the ACM*, Vol 37, No. 9, September 1994, pp. 78-86.

Atre, Shaku, "Selecting End-User Decision Support Tools: Five Key Issues," *Atre Associates Homepage*, 1996, URL: http://www.atre.com/articles/welcome.html

Battershell, A. Lee, "Technology Approach: DoD Versus Boeing (A Comparative Study)," *Acquisition Review Quarterly*, Summer 1995, pp. 213-228.

Baum, David, "The Right Tools for Coding Business Rules," Datamation, March 1, 1995, pp. 36-38.

Barnes, Michael, and David Kelley, "Play by the Rules," Byte, June 1997, pp. 98-102.

Blair, Jim, "Knowledge Management: The Era of Shared Ideas," *A Gartner Group Report: The Future of IT,* Ed. Lynn Morrissey, Vol. 1, No. 1, Forbes Custom

Products, New York City, New York, September 22, 1997, p. 28.

Booch, Grady, *Object-Oriented Analysis and Design with Applications*, 2nd ed., The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1994.

Booch, Grady, "Coming of Age in an Object-Oriented World," *IEEE Software*, November 1994, pp. 33-41.

Burch, John G., *Systems Analysis, Design, and Implementation*, Boyd and Fraser Publishing Company, Boston, Massachusetts, 1992.

Brackett, Michael H., *The Data Warehouse Challenge: Taming Data Chaos*, John Wiley and Sons, Inc., New York, 1996.

Card, David N., and Robert L. Glass, *Measuring Software Design Quality*, Prentice Hall, Englewood Cliffs, New Jersey, 1990.

Chang, Shi-Kuo, *Principles of Visual Programming*, Ed., Prentice Hall, Englewood, New Jersey, 1990.

Cusumano, Michael A., and Richard W. Selby, "How Microsoft Builds Software," Communications of the ACM, Vol. 40, No. 6, June 1997, pp. 53-61.

Danziger, James N. And Kenneth L. Kraemer, *People and Computers: The Impacts of Computing on End Users in Organizations*, Columbia University Press, New York, 1986.

Data Warehousing Institute, *Building, Using, and Managing the Data Warehouse*, Ramon Barquin and Herb Edelstein Ed., Prentice Hall PTR, Upper Saddle River, New Jersey, 1997.

Desfray, Philippe, "Automated Object Design: The Client-Server Case," *IEEE Computer*, February 1996, pp. 62-66.

Evans, Michael W., *The Software Factory: A Fourth Generation Software Engineering Environment*, John Wiley and Sons, New York, 1989.

Fayyad, Usama M., "Data Mining and Knowledge Discovery: Making Sense Out of Data," *IEEE Expert Intelligent Systems & their Applications*, Vol 11, No. 5, October 1996, pp. 20-25.

Fife, Leslie D., "Feature Interaction: How it works in Telecommunications Software," IEEE Potentials, October/November 1996, pp. 35-37.

Fong, Joesph, "Adding a Relational Interface to a Nonrelational Database," *IEEE Software*, Vol. 13, No. 9, September 1996, pp. 89-96.

Genter, Donald R., and Jonathan Grudin, "Design Models for Computer-Human Interfaces," *IEEE Computer*, June 1996, pp. 28-35.

Halfhill, Tom R., "Good-Bye, GUI...Hello, NUI," *Byte*, Vol. 22, No. 7, July 1997.

Harrar, George, "Welcome to IS Boot Camp," *Forbes ASAP*, September 1994, pp. 112-118.

Hertzum, Morten, and Erik Frokjaer, "Browsing and Querying in Online Documentation: A Study of User Interfaces and the Interaction Process," *ACM Transactions on Computer-Human Interaction,* Vol. 3, No. 2, June 1996, pp. 136-161.

Hoffman, Gerald M., *The Technology Payoff: How to Profit with Empowered Workers in the Information Age.* Irwin Professional Publishing, Burr Ridge, Illinois, 1994.

Hohman, Luke, *Journey of the Software Professional: A Sociology of Software Development*, Prentice Hall PTR, Upper Saddle River, New Jersey, 1997.

Jorgensen, Paul C., and Carl Erickson. "Object Oriented Integration Testing," *Communications of the ACM*, Vol. 37, No. 9, September 1994, pp. 30-38.

Kristof, Ray, and Amy Satran, *Interactivity by Design*, Adobe Press, Mountain View, California, July 1995.

Kroenke, David M., *Database Processing: Fundamentals, Design, and Implementation*, 5th ed., Prentice Hall, Englewood Cliffs, New Jersey, 1995.

Kruchten, Philippe, "A Rational Development Process," *Rational Rose Software White Paper*, 1996, URL: http://www.rational.com/

Lasswell, James A., "Innovation Concept-Based Experimentation." *Surface Warfare*, June/July, 1997, pp.32-36.

Lee, Hing-Yan, and Hwee-Leng Ong. "Visualization Support for Data Mining" *IEEE Expert Intelligent Systems & their Applications*, Vol. 11, No. 5, October 1996, pp. 69-75.

Madisetti, Vijay K., "Rapid Digital System Prototyping: Current Practice, Future Challenges," *IEEE Design & Test of Computers,* Fall 1996, pp. 12-22.

MapInfo Consulting Services, *MapInfo and the Data Warehouse*, MapInfo White Paper, MapInfo Corporation, 1996, URL: http://www.mapinfo.com/.

Marko, Bart, "The Future of Financial Management or Paradigm Shifts on the Information Superhighway," *Contract Management*, September 1995, pp. 10-14.

Martin, James, *Application Development Without Programmers*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

Martin, James, *Rapid Application Development*, Macmillan Publishing Company, New York, 1991.

McConnell, Steve C. *Code Complete: A Practical Handbook of Software Construction*, Microsoft Press, Redmond, Washington, 1996.

McConnell, Steve C. *Rapid Development: Taming Wild Software Schedules*. Ed. Jeff Carey. Microsoft Press, Redmond, Wash., 1993.

McConnell, Steve C., "Missing in Action: Information Hiding," *IEEE Software*, Vol. 13, No. 3, March 1996, pp. 127-128.

McConnell, Steve C., "Software's Ten Essentials," *IEEE Software*, Vol. 14, No. 3, March/April 1997, pp. 143-144.

Mesrobian, Edmond, et al. "Mining Geophysical Data for Knowledge," *IEEE Expert Intelligent Systems & their Applications*, Vol 11, No. 5, October 1996, pp. 34-43.

Millington, Don, and Jennifer Stapleton, "Developing a RAD Standard," *IEEE Software*, Vol. 12, No. 9, March 1996, pp. 54-55.

Nemzow, Martin. *Visual Basic Developer's Toolkit Performance Optimization, Rapid Application Development, Debugging, and Distribution*. McGraw-Hill Inc., San Francisco, Calif., 1996.

Neuman, Peter G., "Distributed Systems Have Distributed Risks," *Communications of the ACM*, Vol. 39, No. 11, November 1996, p. 130.

Orfali, Robert, Dan Harkey, and Jeri Edwards. *Essential Client/Server Survival Guide*. 2nd ed., John Wiley & Sons, Inc., New York, 1996.

Poe, Vidette, *Building a Data Warehouse for Decision Support*, Prentice Hall PTR, Upper Saddle River, New Jersey, 1996.

Raden, Neil, "Worlds in Collision," *DBMS online*, August 1997, URL: http://www.dbmsmag.com

Rahmel, Dan, and Ron Rahmel, *Developing Client/Server Applications with Visual Basic 4*. Sams Publishing, Indianapolis, Indiana, 1996.

Reilly, John, P., "Does RAD Live Up to the Hype: Point," *IEEE Software*, September, 1995, p. 24.

Reilly, John, P., *Rapid Prototyping: Moving to Business-Centric Development*, International Thomson Computer Press, Belmont, California, 1996.

Rivas, Erick, "Enterprise Component Modeling Strategies," *Object Magazine*, April 1997, pp. 62-64.

Roof, Larry. *The Revolutionary Guide to Visual Basic 4 Professional*, Ed. John Franklin, Wrox Press Ltd., Chicago, Illinois, 1996.

Rossak, Wilhelm, and et. al., "A Generic Model for Software Architectures," *IEEE Software*, July/August 1997, pp. 84-92.

Rudin, Ken, "Growing Pains," *DBMS online*, February 1997, URL: http://www.dbmsmag.com/9702d14.html

Sadr, Babak, and Patricia J. Dousette, "An OO Project Management Strategy," *IEEE Computer,* September 1996, pp. 33-37.

Sanders G., Lawrence, *Data Modeling*, Boyd and Fraser Publishing Company, San Francisco, California, 1995.

Semich, William J., "Where Do C/S Apps Go Wrong?" *Datamation*, January 21, 1994, pp. 30-36.

Sharp, Oliver, "How to Build Reliable Code," *Byte*, December 1995, pp. 50-51.

Shlaer, Sally, and Stephen J. Mellor, "Recursive Design of an Application-Independent Architecture," *IEEE Software*, Vol. 14, No. 1, January, 1997, pp. 61-72.

Simoudis, Evangelos. "Reality Check for Data Mining," *IEEE Expert Intelligent Systems & their Applications*, Vol. 11, No. 5, October 1996, pp. 26-33.

Simpson, David. "Win95 vs. NT Workstation: You make the call!" *Datamation*, Vol. 42, No. 18, December 1996, pp.110-113.

Stabell, Charles B. "A Decision-oriented Approach to Building DSS" *Building Decision Support Systems*, ed. John L. Bennett, Addison-Wesley, Reading, Massachusetts, 1983.

Stal, Michael, "Software Components: What are They all About?", *Object Magazine*, Vol. 4, No. 7, June 1997, pp. 54-55.

Swaminathan, Venkates, and James Storey, "Domain-Specific Frameworks", *Object Magazine*, April 1997, pp. 53-57.

Taylor, David A. *Object-Oriented Technology: A Manager's Guide*. Addison-Wesley Publishing Company, Menlo Park, Calif., 1990.

Tufte, Edward R. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connecticut, 1983.

Vigder, Mark R., W. Morven Gentleman, and John Dean, "COTS Software Integration: State of the Art," *NRC-CNRC*, Software Engineering Group Report, January 1996.

Webb, Jeff, et. al., *Special Edition Using Visual Basic 4, The Most Complete Reference*. QUE Corporation, Indianapolis, Indiana, 1995.

Wegscheider, Eric, "Toward Code-Free Business Application Development," *IEEE Computer*, March 1997, pp. 35-43.

Williams, John D., "Managing Iteration in OO Projects," *IEEE Computer,* Vol. 13, No. 9, September 1996, pp. 39-43.

Winograd, Terry. "From Programming Environments to Environments for Designing," *Communications of the ACM*, Vol. 38, No. 6, June 1995, pp. 65-74.

Wirfs-Brock, Rebecca, Brian Wilkerson, and Lauren Wiener, *Designing Object-Oriented Software*, Prentice Hall, Englewood Cliffs, New Jersey, 1990.

Yolanda, Reimer J., Ray Ford, and Nicholas P. Wilde, "Merging Task-Centered UI Design with Complex System Development: A Risky Business," *Communications of the ACM*, 1996, pp. 34-42.

Yourdon, Edward, *Structured Design Workshop*, 2nd ed., Yourdon, Inc, New York, 1980.

Yourdon, Edward, "Java, the Web, and Software Development," *IEEE Computer*, August 1996.

Yourdon, Edward, *Death March: The Complete Software Developer's Guide to Surviving 'Mission Impossible' Projects*, Prentice Hall PTR, Upper Saddle River, New Jersey, 1997.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvior, Virginia, 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, California 93943-5101

3. Headquarters, U.S. Army Reserve Command 2
   ATTN: LTC Cleven
   3800 North Camp Creek Parkway SW
   Atlanta, Georgia 30331

4. Professor Daniel Dolk, Code SM/Dk 3
   Department of Systems Management
   Naval Postgraduate School
   555 Dyer Rd Bldg 330
   Monterey, California 93943-5000

5. Professor James Emery, Code O5 1
   Department of Systems Management
   Naval Postgraduate School
   555 Dyer Rd Bldg 330
   Monterey, California 93943-5000

6. LCDR Dale Courtney, Code O5C 1
   Department of Systems Management
   Naval Postgraduate School
   555 Dyer Rd Bldg 330
   Monterey, California 93943-5000

7. Professor George Thomas, Code SM/Te 1
   Department of Systems Management
   Naval Postgraduate School
   555 Dyer Rd Bldg 330
   Monterey, California 93943-5000

8. LCDR Peter R. Falk 3
   1009 Timberwood Court
   Virginia Beach, Virginia 23454